ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Σημειώσεις του μαθήματος

ΕΦΑΡΜΟΓΕΣ ΘΕΩΡΙΑΣ ΓΡΑΦΗΜΑΤΩΝ

Κ. Μανές - Ι. Τασούλας

ΠΕΙΡΑΙΑΣ 2023

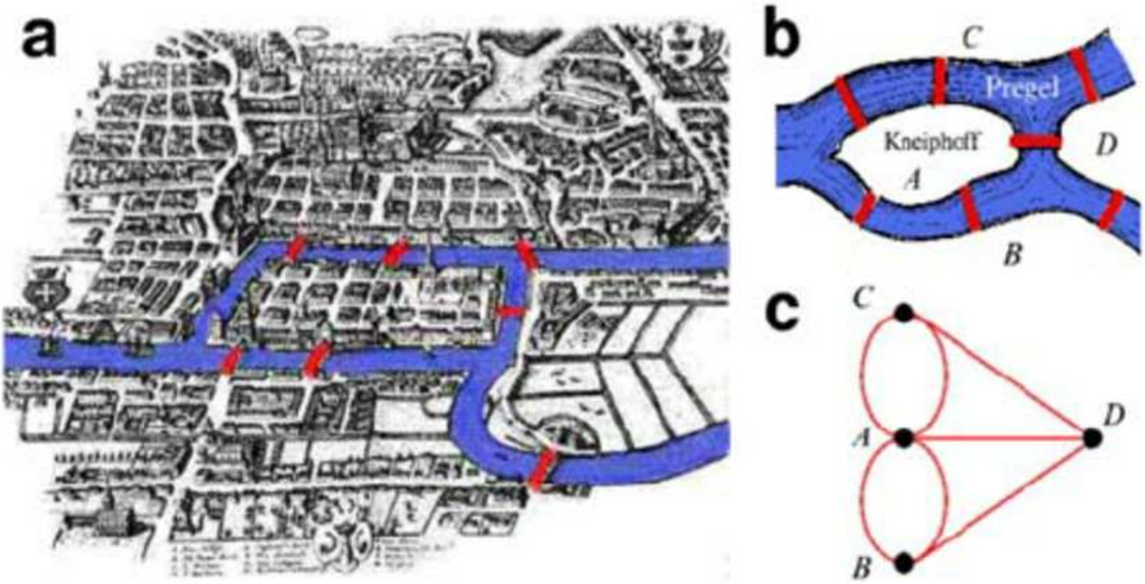
Περιεχόμενα

Οι παρούσες σημειώσεις, βασίζονται σε προηγούμενες σημειώσεις του μαθήματος που έχουν συγγράψει ο Καθηγητής κ. Αριστείδης Σαπουνάκης και ο Καθηγητής κ. Παναγιώτης-Γεώργιος Τσικούρας.

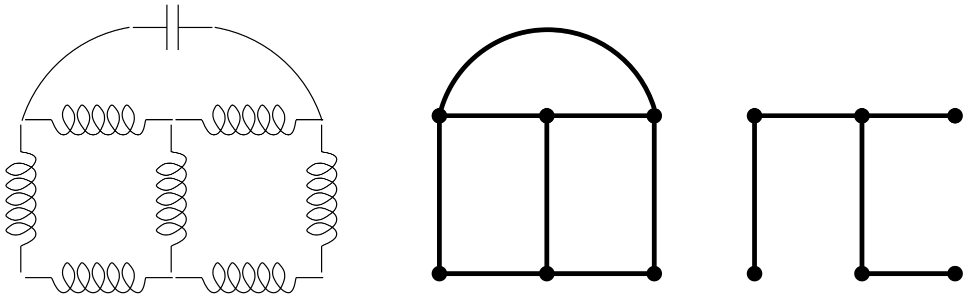
ΕΙΣΑΓΩΓΗ - ΙΣΤΟΡΙΚΟ

Euler (1736): Γέφυρες του Königsberg

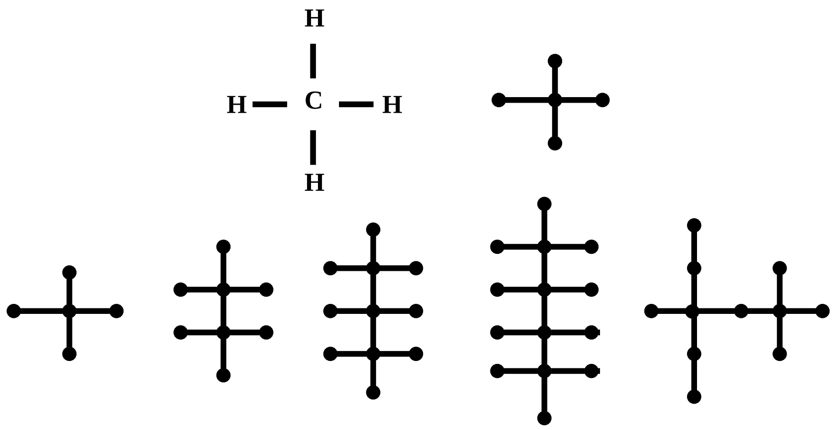
Μπορεί κάποιος να περάσει ακριβώς μια φορά από κάθε γέφυρα;



Kirchhoff (1847): Γενετικό δένδρο



Cayley (1857): Πλήθος κορεσμένων υδρογονάνθρακων C_nH_{2n+2}



Μερικές από τις εφαρμογές της Θεωρίας Γραφημάτων:

Πληροφορική (Δένδρα, Δυαδικά δένδρα, Διατεταγμένα δένδρα, Διάτρεξη (διάσχιση) δένδρων, Προγραμματισμός, Συνδεσμολογία κ.λπ.).

Αλγόριθμοι (Αλγόριθμοι γραφημάτων, Αναζήτηση πρώτα κατά πλάτος, Αναζήτηση πρώτα κατά βάθος, Τοπολογική διάταξη, Κατάταξη έργων με προθεσμίες κ.λπ.).

Διοίκηση Επιχειρήσεων (Οργανογράμματα, Κεντρικά σημεία κ.λπ.).

Οδοποιΐα (Οδικά δίκτυα - χωρητικότητα - μέγιστη ροή, Σηματοδότηση δρόμων).

Υδραυλικά (Δίκτυα - χωρητικότητα -μέγιστη ροή).

Ιστορία - Κοινωνιολογία (Γενεαλογικά δένδρα, Φιλία (γραφήματα δεσμών), Έρωτας (γραφήματα τόξων)).

ΠΡΩΤΟ ΜΕΡΟΣ: ΕΙΣΑΓΩΓΗ ΣΤΑ ΓΡΑΦΗΜΑΤΑ ΔΕΣΜΩΝ

1. ΒΑΣΙΚΟΙ ΟΡΙΣΜΟΙ

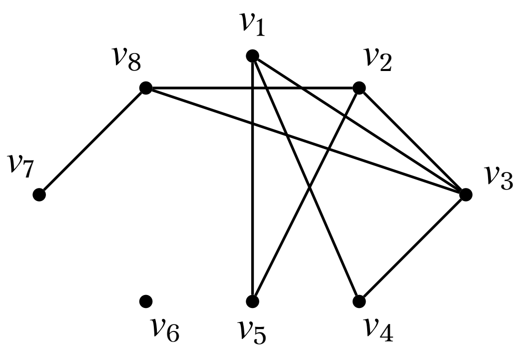
Κάθε δυάδα $G = (V(G), E(G))$, ή (V, E), ή (X, E) όπου V είναι ένα μη κενό σύνολο και E είναι ένα σύνολο από (μη διατεταγμένα) ζεύγη $\{u, v\}$, $u, v \in V$ ονομάζεται **γράφημα δεσμών** (graph), ή **απροσανατόλιστο γράφημα** (undirected graph).

Τα στοιχεία του V καλούνται **κορυφές**, ή **σημεία**, ή **κόμβοι** (vertices, nodes, points), ενώ τα στοιχεία του E καλούνται **δεσμοί**, ή **γραμμές**, ή **χορδές**, ή **πλευρές**, ή **ακμές** (edges, lines).

Θα ασχοληθούμε εδώ με **πεπερασμένα γραφήματα**, (δηλαδή $|V| \in \mathbb{N}^*$). Το E μπορεί να είναι $\emptyset$. Συχνά γράφουμε $|V| = p$ ή n και $|E| = q$. Ο πληθάριθμος $|V|$ ονομάζεται **τάξη** (order) του γραφήματος, ενώ ο πληθάριθμος $|E|$ ονομάζεται **μέγεθος** (size) του γραφήματος.

Αν $\{u, v\} \in E$, λέμε ότι τα u, v είναι **άκρα** του δεσμού $\{u, v\}$ ή, ισοδύναμα, ότι το u (και το v) **καλύπτει** τον δεσμό $\{u, v\}$. Αν δύο δεσμοί έχουν κοινή μια κορυφή, λέμε ότι είναι **γειτονικοί**.

Παράδειγμα: Η δυάδα $G = (V, E)$ όπου $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$ και $E = \{\{v_1, v_3\}, \{v_1, v_4\}, \{v_1, v_5\}, \{v_2, v_3\}, \{v_2, v_5\}, \{v_2, v_8\}, \{v_3, v_4\}, \{v_3, v_8\}, \{v_7, v_8\}\}$ είναι ένα γράφημα δεσμών. Η γραφική του απεικόνιση είναι η ακόλουθη:



Αν οι u, v ταυτίζονται έχουμε ένα **βρόχο**.

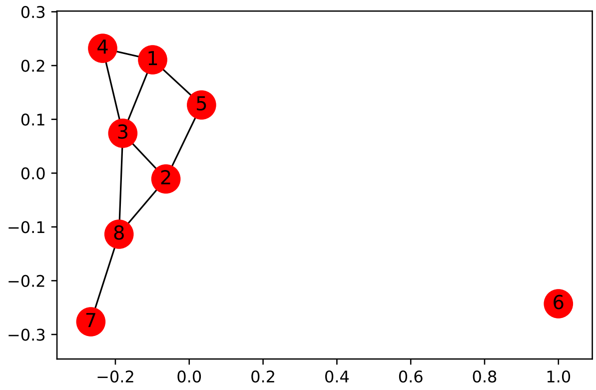
Παρατήρηση: Δεδομένου ότι σε ένα σύνολο επιτρέπεται μία μόνο εμφάνιση κάθε στοιχείου του, από τον ορισμό του γραφήματος δεσμών προκύπτει ότι σε αυτό δεν επιτρέπονται ούτε βρόχοι, ούτε πολλαπλοί δεσμοί που να συνδέουν το ίδιο ζεύγος κορυφών. Τα γραφήματα αυτά ονομάζονται **απλά γραφήματα** (simple graphs) και με τέτοια θα ασχοληθούμε, εκτός αν αναφερθεί ρητά το αντίθετο.

Στο επόμενο πρόγραμμα χρησιμοποιούμε την βιβλιοθήκη networkx της Python για να ορίσουμε το γράφημα G του πρώτου παραδείγματος.

```
import networkx as nx
import matplotlib.pyplot as plt

G = nx.Graph() #Create an empty graph

V = [1,2,3,4,5,6,7,8] #V is the set of vertices of G
E = [[1,3],[1,4],[1,5],[2,3],[2,5],[2,8],[3,4],[3,8],[7,8]] #E is the set of edges of
    G

G.add_nodes_from(V)
G.add_edges_from(E)

print("G has order |V(G)|=",G.order(),"and size |E(G)|=",G.size())
print("V(G):",G.nodes()) #Print the nodes of G
print("E(G):", G.edges()) #Print the edges of G
for v in G:
    print("The neighbors of", v, "are:", list(G.neighbors(v)))

nx.draw_networkx(G) #Draw the graph G
plt.savefig("lect01a.eps") #Save the drawing of G
plt.show() #Show the drawing of G on screen
```

Output:

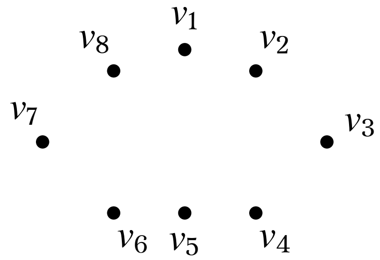
```
G has order |V(G)|= 8 and size |E(G)|= 9
V(G): [1, 2, 3, 4, 5, 6, 7, 8]
E(G): [(1, 3), (1, 4), (1, 5), (2, 3), (2, 5), (2, 8), (3, 4), (3, 8), (7, 8)]
The neighbors of 1 are: [3, 4, 5]
The neighbors of 2 are: [3, 5, 8]
The neighbors of 3 are: [1, 2, 4, 8]
The neighbors of 4 are: [1, 3]
The neighbors of 5 are: [1, 2]
The neighbors of 6 are: []
The neighbors of 7 are: [8]
The neighbors of 8 are: [2, 3, 7]
```



ΜΟΡΦΕΣ ΓΡΑΦΗΜΑΤΩΝ

1) **Μηδενικό γράφημα**: $G = (V, E)$ με $E = \emptyset$.

Παράδειγμα:

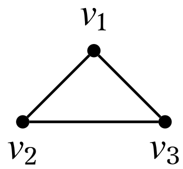


2) **Τετριμμένο γράφημα**: $G = (V, E)$ με $|V| = 1$.

3) **Πλήρες γράφημα**: $G = (V, E)$ τέτοιο ώστε για κάθε $u, v \in V$ με $u \neq v$ ισχύει ότι $\{u, v\} \in E$. Το πλήρες γράφημα με n κορυφές συμβολίζεται με K_n.

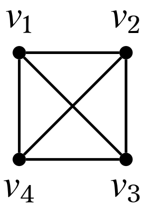
Παράδειγμα: Το γράφημα K_3 είναι το:

K_3



ενώ το γράφημα K_4 είναι το :

K_4



Παρατήρηση: Το K_n έχει n κορυφές και $\binom{n}{2}$ δεσμούς (όσα και τα ζευγάρια του $[n]$).

Στην βιβλιοθήκη networkx το πλήρες γράφημα με n κορυφές κατασκευάζεται χρησιμοποιώντας την μέθοδο `complete_graph(n)`, ή χρησιμοποιώντας τις επόμενες εντολές:

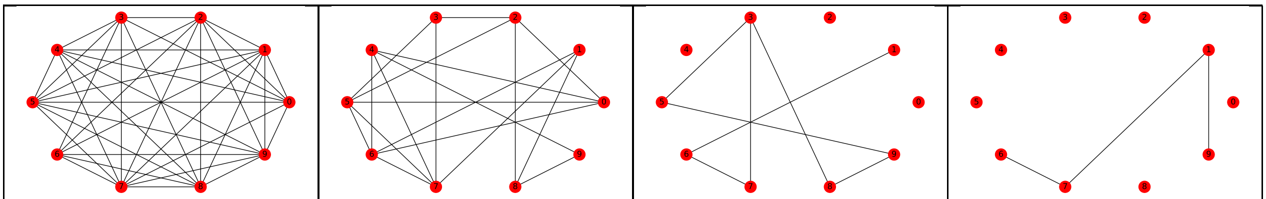
```python
import networkx as nx
import matplotlib.pyplot as plt

n = 7 #number of vertices

Kn = nx.complete_graph(n)
nx.draw_circular(Kn,with_labels=True)
plt.show()

Kn = nx.Graph()
Kn.add_nodes_from(range(1,n+1))
for i in range(1,n+1):
    for j in range(i+1,n+1):
        Kn.add_edge(i,j)
nx.draw_circular(Kn,with_labels=True)
plt.show()
```

4) **Τυχαίο γράφημα**: Κάποιες φορές καλούμαστε να δοκιμάσουμε αλγορίθμους ή ιδέες μας πάνω σε διάφορα παραδείγματα γραφημάτων. Μπορούμε να φτιάχνουμε τέτοια "τυχαία" παραδείγματα χρησιμοποιώντας έτοιμες μεθόδους της βιβλιοθήκης networkx ή γράφοντας δικές μας μεθόδους, όπως στα επόμενα παραδείγματα.

Η πιο απλή ιδέα κατασκευής ενός τυχαίου γραφήματος με n κορυφές είναι το **μοντέλο των Erdős - Renyi** όπου για κάθε ζεύγος κορυφών επιλέγουμε να δημιουργήσουμε τον δεσμό που τις συνδέει με πιθανότητα p.

Ένα τέτοιο γράφημα προκύπτει χρησιμοποιώντας την μέθοδο `gnp_random_graph(n,p)`:

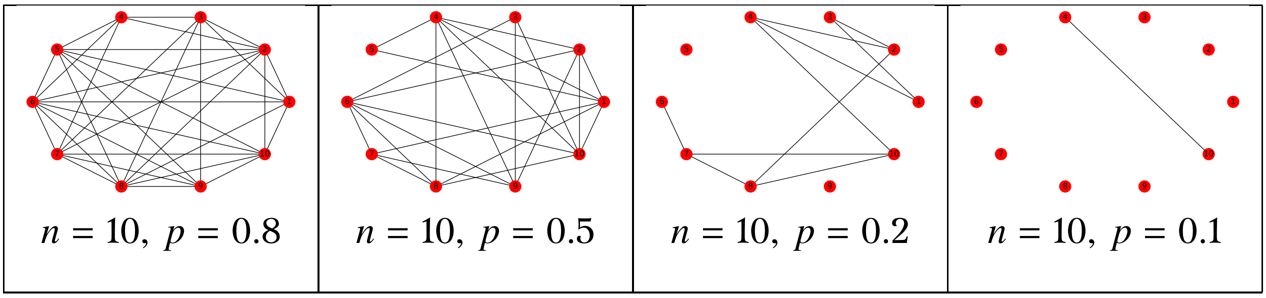
```
import networkx as nx
import matplotlib.pyplot as plt

def random_gnp_graph(n,p,name):
    R = nx.gnp_random_graph(n,p)
    nx.draw_circular(R,with_labels=True)
    plt.savefig(name+".eps")
    plt.show()

random_gnp_graph(10,0.8,"R1")
random_gnp_graph(10,0.5,"R2")
random_gnp_graph(10,0.2,"R3")
random_gnp_graph(10,0.1,"R4")
```



$n = 10,\ p = 0.8$ | $n = 10,\ p = 0.5$ | $n = 10,\ p = 0.2$ | $n = 10,\ p = 0.1$

Μια απλή υλοποίηση της μεθόδου `gnp_random_graph(n,p)`:

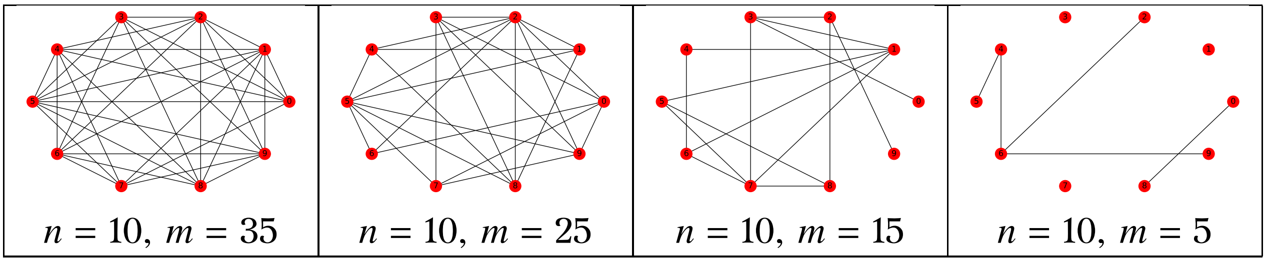
```
import networkx as nx
import matplotlib.pyplot as plt
import random #random numbers

def random_gnp_graph2(n,p,name):
    R = nx.Graph()
    R.add_nodes_from(range(1,n+1))
    for i in range(1,n+1):
        for j in range(i+1,n+1):
            if random.uniform(0,1) <= p:
                R.add_edge(i,j)
    nx.draw_circular(R,with_labels=True)
    plt.savefig(name+".eps")
    plt.show()

random_gnp_graph2(10,0.8,"R9")
random_gnp_graph2(10,0.5,"R10")
random_gnp_graph2(10,0.2,"R11")
random_gnp_graph2(10,0.1,"R12")
```



Παρατήρηση: Επειδή κάθε ένας από τους $\binom{n}{2}$ πιθανούς δεσμούς επιλέγεται με πιθανότητα p έπεται ότι το τυχαίο γράφημα που προκύπτει κατά μέσο όρο αναμένεται να έχει $p\binom{n}{2}$ δεσμούς.

Στην περίπτωση όπου θέλουμε το τυχαίο γράφημα να έχει n κορυφές και ακριβώς m δεσμούς μπορούμε να χρησιμοποιήσουμε την μέθοδο `gnm_random_graph(n,m)`:

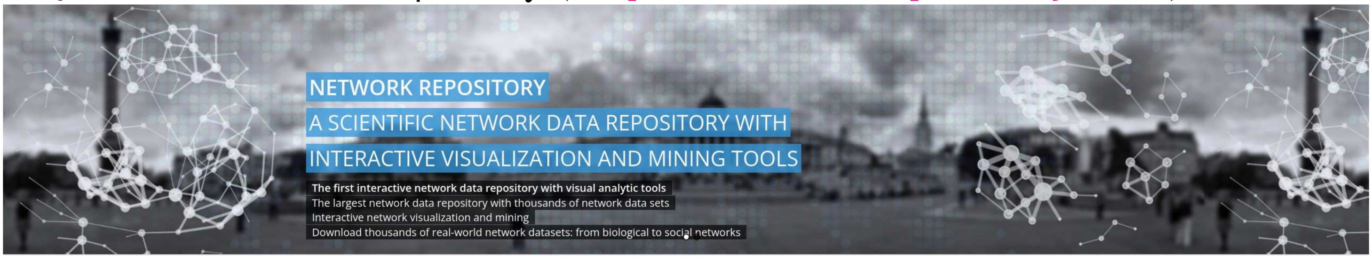
```python
import networkx as nx
import matplotlib.pyplot as plt

def random_gnm_graph(n,m,name):
    R = nx.gnm_random_graph(n,m) #0 <= m <= n(n-1)/2
    nx.draw_circular(R,with_labels=True)
    plt.savefig(name+".eps")
    plt.show()

random_gnm_graph(10,35,"R5")
random_gnm_graph(10,25,"R6")
random_gnm_graph(10,15,"R7")
random_gnm_graph(10,5,"R8")
```

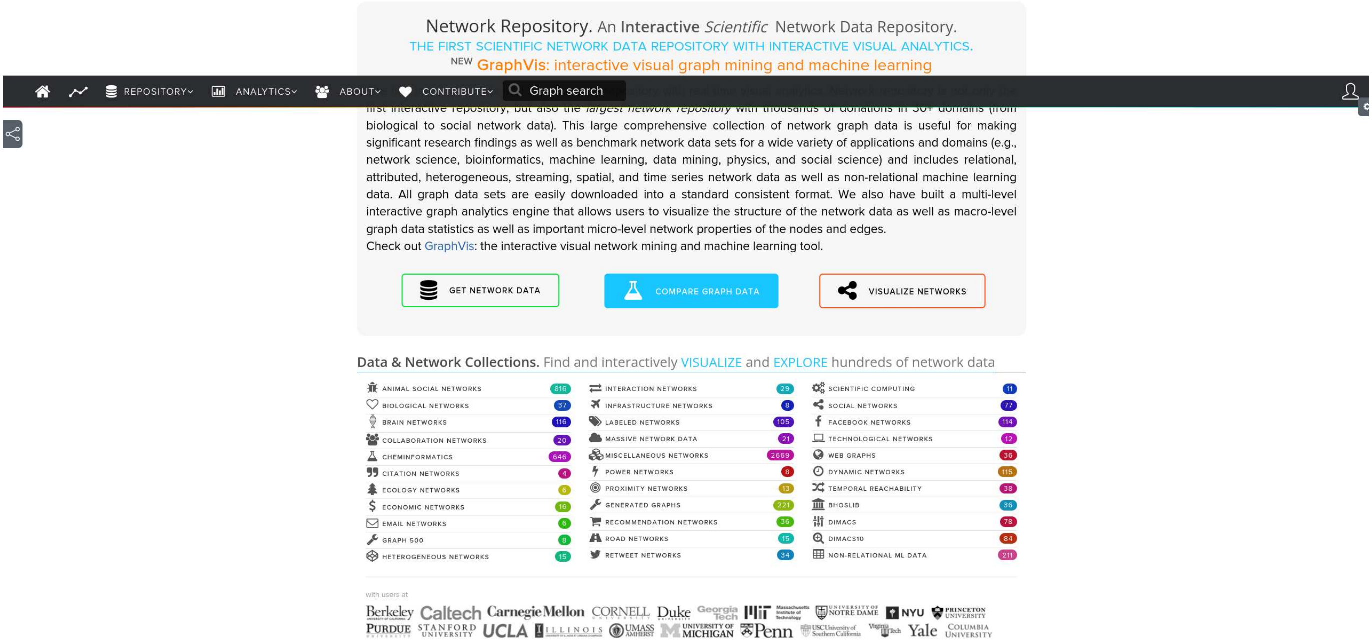


Παρατήρηση (*): Μια υλοποίηση της μεθόδου `gnm_random_graph(n,m)` μπορεί να γίνει κατασκευάζοντας έναν τυχαίο υποσύνολο του $[\binom{n}{2}]$ με m στοιχεία.

ΠΗΓΕΣ ΑΝΟΙΧΤΩΝ ΔΕΔΟΜΕΝΩΝ ΓΙΑ ΓΡΑΦΗΜΑΤΑ

Υπάρχουν αρκετοί ιστότοποι με συλλογές ανοιχτών δεδομένων που αφορούν γραφήματα τα οποία εμφανίζονται σε πραγματικές καταστάσεις. Ένας τέτοιος ιστότοπος είναι το Network Repository (https://networkrepository.com/):





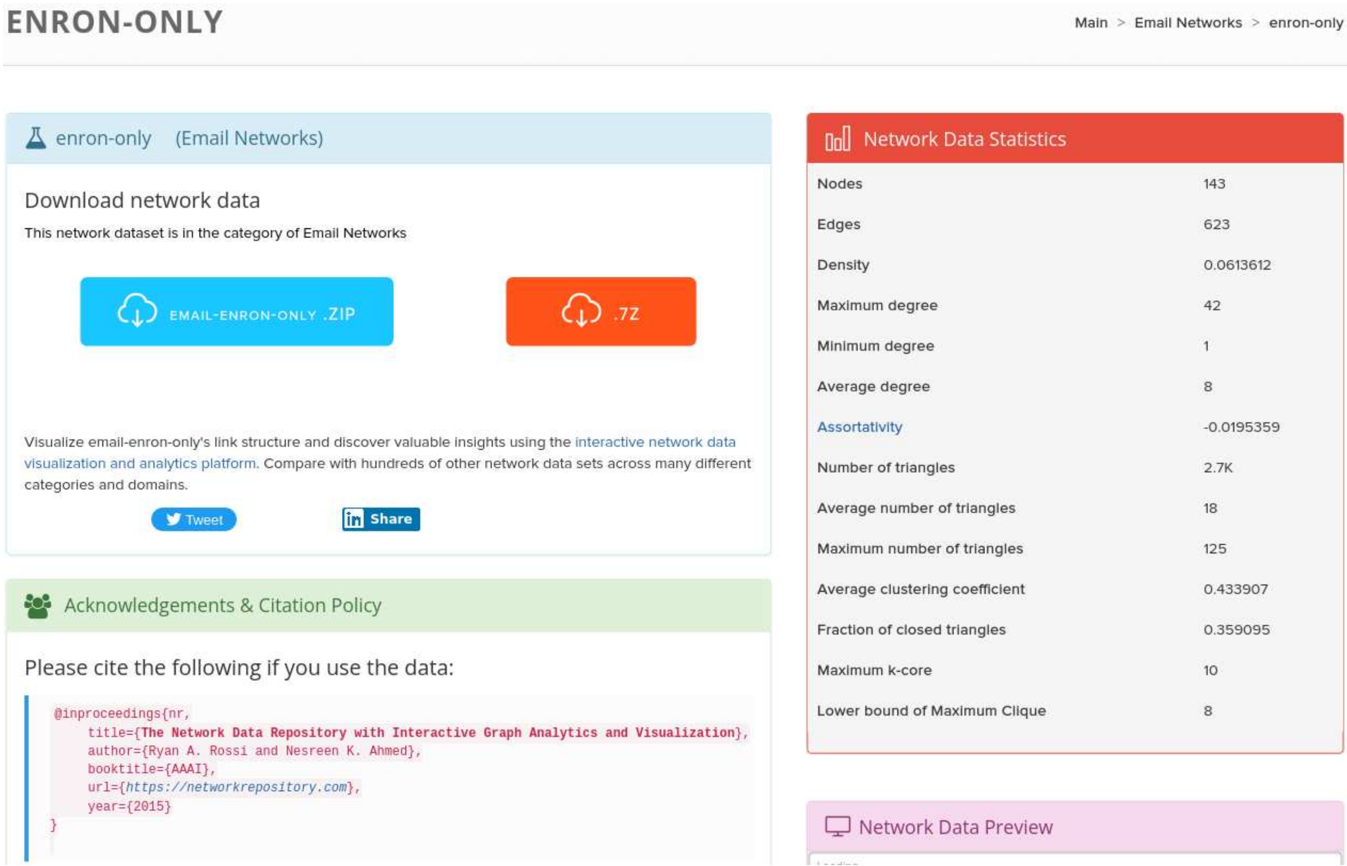
Συνήθως, τα δεδομένα που αφορούν τα γραφήματα είναι διαθέσιμα σε μορφή αρχείου κειμένου που περιέχει λίστα με δεσμούς του γραφήματος (δύο αριθμοί σε κάθε γραμμή, οι αριθμοί δηλώνουν τις ετικέτες των κορυφών που ενώνει ο δεσμός).

Η βιβλιοθήκη `networkx` έχει την μέθοδο `read_edgelist('filename.edges')` για την ανάγνωση του γραφήματος από το αρχείο `filename.edges` το οποίο περιέχει την λίστα των δεσμών του γραφήματος.

```
import networkx as nx
import matplotlib.pyplot as plt

G = nx.read_edgelist('realgraphs/email-EU.edges')

print("Number of nodes:", G.order(), "Number of edges:", G.size())
```

Output:

```
Number of nodes: 32430 Number of edges: 54397
```

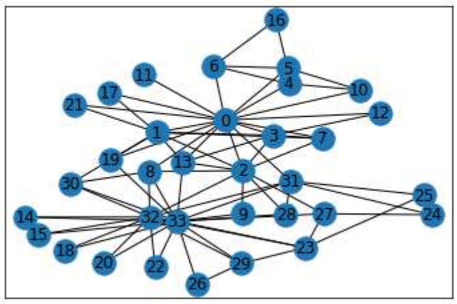


Σχήμα 1. Παράδειγμα από το Network Repository

Επίσης, αρκετά δημοφιλής είναι ο τύπος αρχείου `mtx` (matrix market file) το οποίο εκτός από την λίστα των δεσμών περιέχει το πλήθος των δεσμών και τον συνολικό αριθμό των κορυφών, επίσης μπορεί να περιέχει σχόλια. Για την αναγνωση αρχείων `mtx` μπορεί να χρησιμοοποιηθεί η μέθοδος `mmread('filename.mtx')` της βιβλιοθήκης `scipy.io`.

```python
import networkx as nx
import matplotlib.pyplot as plt
from scipy.io import mmread

a = mmread('realgraphs/soc-karate.mtx')
G = nx.Graph(a)

pos = nx.layout.kamada_kawai_layout(G)
nx.draw_networkx(G,pos)

plt.show()
```

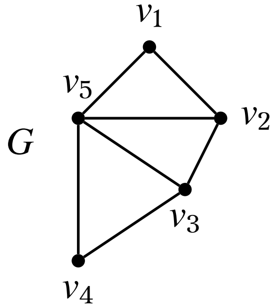
Output:



ΥΠΟΓΡΑΦΗΜΑΤΑ

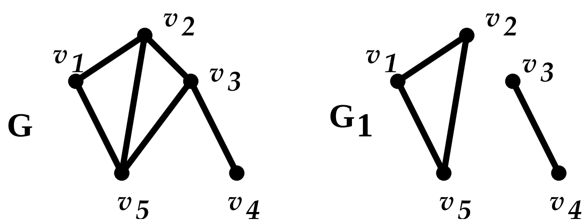
1) **Υπογράφημα** (subgraph) του $G = (V, E)$: Ένα γράφημα $G_1 = (V_1, E_1)$ με $V_1 \subseteq V$ και $E_1 \subseteq E$.

Παράδειγμα:



2) **Γενετικό** (ή **γεννητικό**, ή **μερικό**) **γράφημα**, ή **γράφημα ζεύξης** (spanning graph) του $G = (V, E)$: Ένα γράφημα $G_1 = (V, E_1)$ με $E_1 \subseteq E$.

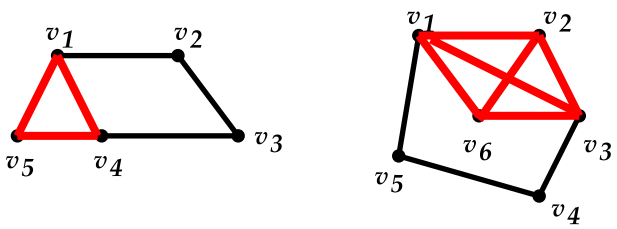
Παράδειγμα:



3) **Κλίκα** (clique): Κάθε πλήρες υπογράφημα του G.

Μέγιστη κλίκα: Κλίκα με το μέγιστο δυνατό αριθμό κόμβων.

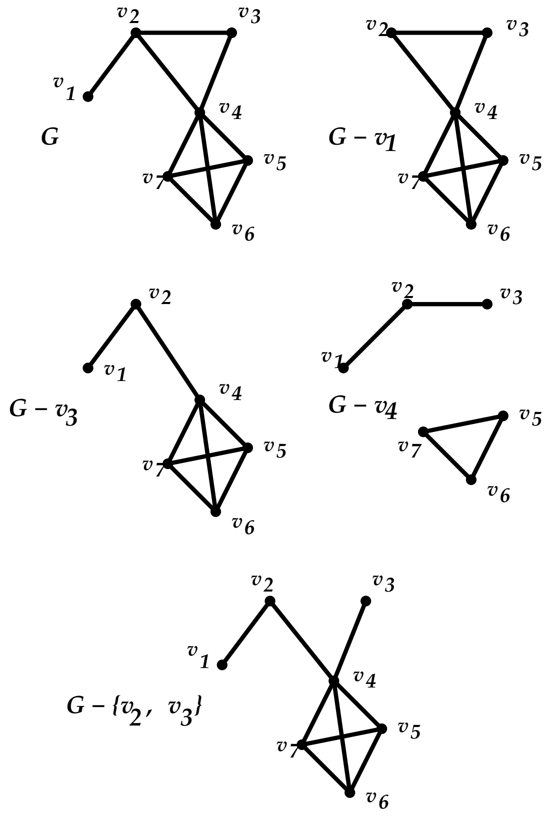
Παραδείγματα:



Οι μέγιστες κλίκες των δύο παραπάνω γραφημάτων είναι οι K_3 και K_4 αντίστοιχα.

4) Αν $G=(V,E)$ και $v \in V$, $e \in E$ ορίζουμε τα υπογραφήματα $\mathbf{G-v}$, $\mathbf{G-e}$ ως εξής:
$V(G-v) = V \setminus \{v\}$, $E(G-v) = E \setminus \{e_i \in E : v \in e_i\}$, ενώ
$V(G-e) = V$, $E(G-e) = E \setminus \{e\}$.

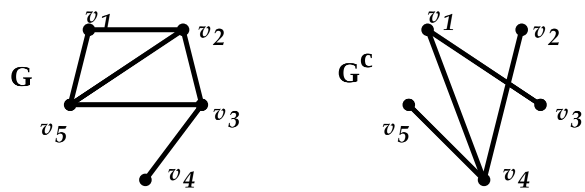
Παραδείγματα:



ΣΥΜΠΛΗΡΩΜΑ

Συμπλήρωμα (complement) G^c (ή $\overline{G}$) του $G=(V,E)$ με $|V|=n$ είναι ένα γράφημα $G^c=(V,E^c)$, με $E^c = E(K_n) \setminus E(G)$.

Παράδειγμα:



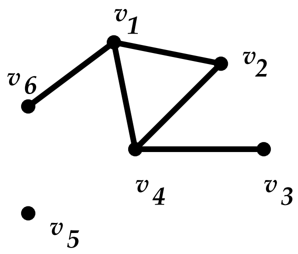
ΒΑΘΜΟΣ

Για κάθε $v \in V$ ορίζουμε $\Gamma_G(v) = \{u \in V(G) : \{v, u\} \in E(G)\}$.

Τότε $d_G(v)$, ή $d(v)$, ή $\deg(v) = |\Gamma_G(v)|$, είναι ο **βαθμός** (degree) του κόμβου v.

Δηλαδή, βαθμός του v στο G, λέγεται το πλήθος των δεσμών του G των οποίων ο v είναι άκρο, ή ισοδύναμα το πλήθος των γειτόνων του v.

Παράδειγμα:



Στο παραπάνω γράφημα, οι κόμβοι του έχουν τους ακόλουθους βαθμούς:

$$\begin{aligned}
&d(v_1) = d(v_4) = 3,\\
&d(v_2) = 2,\\
&d(v_3) = d(v_6) = 1,\\
&d(v_5) = 0.
\end{aligned}$$

Κάθε κόμβος βαθμού μηδέν λέγεται **μεμονωμένος** κόμβος.

Ένα γράφημα G λέγεται d-**κανονικό** αν $d_G(v) = d$, για κάθε $v \in V$.

Ο **ελάχιστος** (αντ. **μέγιστος**) **βαθμός** των κορυφών ενός γραφήματος G θα συμβολίζεται με $\delta(G)$ (αντ. $\Delta(G)$).

Έστω $G = (V, E)$ με $V = \{v_1, v_2, \ldots, v_n\}$.

Ακολουθία βαθμών του G λέγεται η πεπερασμένη ακολουθία

$$(d(v_1), d(v_2), \ldots, d(v_n)).$$

Παράδειγμα: Η ακολουθία βαθμών του παραπάνω γραφήματος είναι (3,3,2,1,1,0).
Παρατήρηση: Συνήθως γράφουμε την ακολουθία βαθμών ενός γραφήματος σε φθίνουσα σειρά.

```
import networkx as nx
import matplotlib.pyplot as plt

G = nx.Graph() #Create an empty graph

V = [1,2,3,4,5,6] #V is the set of vertices of G
E = [[1,2],[1,4],[1,6],[2,4],[3,4]] #E is the set of edges of G

G.add_nodes_from(V)
G.add_edges_from(E)

DegreeSeq = []
for v in G.nodes:
    DegreeSeq.append(G.degree(v))
    print("Vertex",v,"has degree",G.degree(v))
DegreeSeq.sort(reverse=True)
print("Degree sequence of G:", DegreeSeq)
```

Output:

```
Vertex 1 has degree 3
Vertex 2 has degree 2
Vertex 3 has degree 1
Vertex 4 has degree 3
Vertex 5 has degree 0
Vertex 6 has degree 1
Degree sequence of G: [3, 3, 2, 1, 1, 0]
```

Οι ακολουθίες βαθμών έχουν ορισμένους περιορισμούς, δηλαδή δεν αντιστοιχούν όλες οι ακολουθίες φυσικών αριθμών σε ακολουθίες βαθμών γραφημάτων. Για παράδειγμα, δεν υπάρχει γράφημα με 5 κορυφές το οποίο έχει ακολουθία βαθμών $(6, 4, 4, 4, 4)$ διότι σε κάθε γράφημα G με 5 κορυφές ισχύει ότι $\Delta(G) \leq 4$.

Μια ακολουθία φυσικών αριθμών $(d_1, d_2, \ldots, d_n)$ ονομάζεται **γραφική** (graphical) αν υπάρχει γράφημα G με ακολουθία βαθμών την ακολουθία $(d_1, d_2, \ldots, d_n)$.

Η μηδενική ακολουθία $(0, 0, \ldots, 0)$ μήκους n αντιστοιχεί στο μηδενικό γράφημα με n κορυφές.

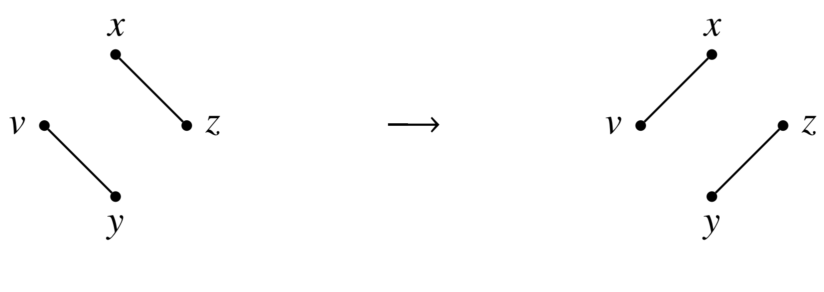
Η επόμενη πρόταση δίνει μια αναγκαία και ικανή συνθήκη για το πότε μια ακολουθία είναι γραφική.

Πρόταση 1 (Θεώρημα Havel - Hakimi).
Η φθίνουσα ακολουθία $(d_1, d_2, \ldots, d_n)$ *είναι γραφική αν και μόνο αν η ακολουθία* $(d_2 - 1, d_3 - 1, \ldots, d_{d_1+1} - 1, d_{d_1+2}, d_{d_1+3}, \ldots, d_n)$ *είναι γραφική.*

Απόδειξη. Το αντίστροφο είναι προφανές. Πράγματι, δοθέντος ενός γραφήματος με ακολουθία βαθμών d' προσθέτουμε σε αυτό μια νέα κορυφή v η οποία ενώνεται με τις d_1 κορυφές με τον μεγαλύτερο βαθμό, παίρνοντας έτσι ένα γράφημα με ακολουθία βαθμών d.

Για το ευθύ, θεωρούμε ένα γράφημα $G = (V, E)$ με ακολουθία βαθμών d και μια οποιαδήποτε κορυφή v, έστω βαθμού k. Η απόδειξη βασίζεται στην παρατήρηση ότι μπορούμε να εναλλάξουμε τους δεσμούς του G χωρίς να αλλάξουν οι βαθμοί των κορυφών του, έτσι ώστε η v να συνδέεται με τις (υπόλοιπες) κορυφές που έχουν τους k μεγαλύτερους βαθμούς.

Πράγματι, έστω ότι $x, y \in V$ με $d(x) > d(y)$ και η v έχει γείτονα τον y αλλά όχι τον x.



Επειδή $d(x) > d(y)$ υπάρχει γείτονας z της x που δεν είναι γείτονας της y.

Διαγράφοντας τις ακμές $\{v, y\}$ και $\{x, z\}$ και προσθέτοντας τις ακμές $\{v, x\}$ και $\{y, z\}$ προκύπτει ένα γράφημα G' στο οποίο οι κορυφές διατηρούν τους ίδιους βαθμούς που είχαν στο G και η v συνδέεται με την x.

Επαναλαμβάνοντας αυτόν τον μετασχηματισμό μπορούμε να πετύχουμε την ζητούμενη ιδιότητα.

Τέλος, διαγράφοντας την κορυφή v με τον μέγιστο βαθμό παίρνουμε ένα γράφημα με ακολουθία βαθμών την d'. □

Παράδειγμα : Σύμφωνα με το θεώρημα Havel - Hakimi η ακολουθία $(6, 6, 4, 4, 2, 2, 2, 2)$ είναι γραφική αν και μόνο αν η ακολουθία

$$(6-1, 4-1, 4-1, 2-1, 2-1, 2-1, 2) = (5, 3, 3, 1, 1, 1, 2)$$

είναι γραφική. Η ακολουθία $(5, 3, 3, 2, 1, 1, 1)$ είναι γραφική ανν η ακολουθία

$$(3-1, 3-1, 2-1, 1-1, 1-1, 1) = (2, 2, 1, 0, 0, 1)$$

είναι γραφική. Η ακολουθία $(2, 2, 1, 1, 0, 0)$ είναι γραφική ανν η ακολουθία

$$(2-1, 1-1, 1, 0, 0) = (1, 0, 1, 0, 0)$$

είναι γραφική. Η ακολουθία $(1, 1, 0, 0, 0)$ είναι πράγματι γραφική, αφού αντιστοιχεί στο γράφημα,

άρα και η ακολουθία $(6, 6, 4, 4, 2, 2, 2, 2)$ είναι επίσης γραφική.

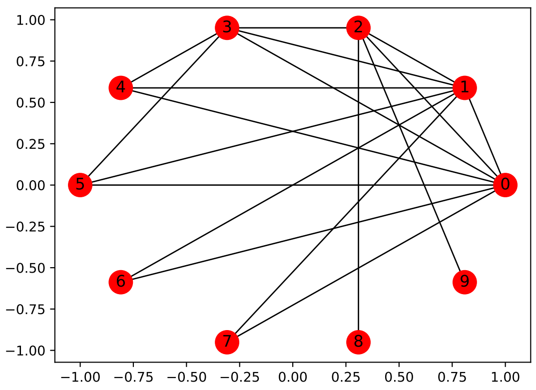
Το θεώρημα Havel - Hakimi δίνει ένα αναδρομικό κριτήριο για τον έλεγχο του κατά πόσο μια ακολουθία είναι γραφική, και ανάγει το πρόβλημα στον έλεγχο μια ακολουθίας με μήκος ένα λιγότερο από την αρχική. Μπορούμε να εφαρμόσουμε ξανά το θεώρημα στην ακολουθία $(d_2-1, d_3-1, \ldots, d_{d_1+1}-1, d_{d_1+2}, d_{d_1+3}, \ldots, d_n)$ (αρκεί πρώτα να την διατάξουμε σε φθίνουσα σειρά) και να προκύψει μια ακολουθία με μικρότερο μήκος, μέχρις ότου να καταλήξουμε σε μία ακολουθία για την οποία μπορούμε εύκολα να συμπεράνουμε αν είναι γραφική ή όχι. Οι ακολουθίες που προκύπτουν κατά την αναδρομική εφαρμογή του θεωρήματος Havel - Hakimi είτε είναι όλες γραφικές είτε καμία γραφική.

Για τον έλεγχο ύπαρξης και κατασκευής ενός γραφήματος με συγκεκριμένη ακολουθία βαθμών μπορούμε να χρησιμοποιήσουμε τις μεθόδους `is_graphical` και `havel_hakimi_graph` της βιβλιοθήκης networkx.

```python
import networkx as nx
import matplotlib.pyplot as plt

def Graph_Check(Seq):
    if nx.is_graphical(Seq):
        print("The sequence",Seq,"is graphical")
        G = nx.havel_hakimi_graph(Seq)
        nx.draw_circular(G,with_labels=True)
        plt.show()
    else:
        print("The sequence",Seq,"is not graphical")
    print("")

Seq1 = [7,7,5,5,3,3,2,2,1,1]
Seq2 = [7,7,7,3,3,2,1,1,1]

Graph_Check(Seq1)
Graph_Check(Seq2)
```

Output:

```
The sequence [7, 7, 5, 5, 3, 3, 2, 2, 1, 1] is graphical
```




```
The sequence [7, 7, 7, 3, 3, 2, 1, 1, 1] is not graphical
```

Ο ΑΛΓΟΡΙΘΜΟΣ ΤΩΝ HAVEL - HAKIMI

Επίσης, το θεώρημα Havel - Hakimi μας προσφέρει μια απλή επαναληπτική μέθοδο για να κατασκευάζουμε γραφήματα με συγκεκριμένη ακολουθία βαθμών.

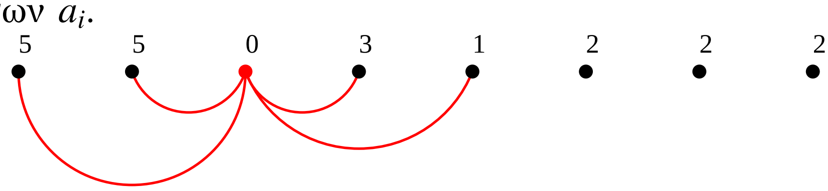
Συγκεκριμένα, αν $(d_1, d_2, \ldots, d_n)$ είναι μια γραφική ακολουθία ο αλγόριθμος κατασκευής χρησιμοποιεί μια βοηθητική ακολουθία $(a_1, a_2, \ldots, a_n)$ και λειτουργεί ως εξής:

Αρχικά θεωρούμε n κορυφές $\{v_1, v_2, \ldots, v_n\}$ βαθμού 0 και για κάθε κορυφή σημειώνουμε τον αριθμό a_i των δεσμών που απαιτούνται για να αποκτήσει βαθμό d_i.

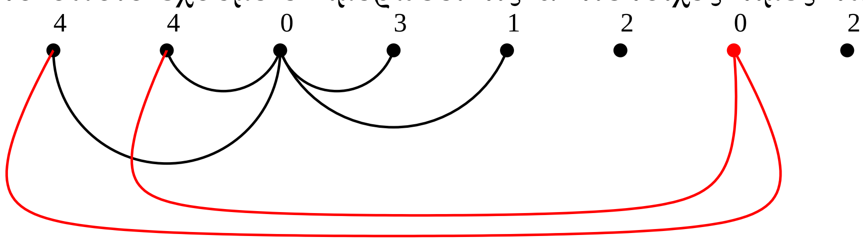
Σε κάθε βήμα επιλέγουμε οποιαδήποτε κορυφή, έστω την v_k, με $a_k > 0$, και την συνδέουμε με a_k σε πλήθος κορυφές που έχουν τις μεγαλύτερες δυνατές θετικές τιμές στην ακολουθία $(a_1, a_2, \ldots, a_n)$. Ενημερώνουμε την ακολουθία $(a_1, a_2, \ldots, a_k)$ και επαναλαμβάνουμε το βήμα αυτό μέχρις ότου όλα τα a_k γίνουν μηδενικά.

Παράδειγμα : Για να κατασκευάσουμε ένα γράφημα με ακολουθία βαθμών $(6, 6, 4, 4, 2, 2, 2, 2)$ αρχικά θεωρούμε 8 κορυφές βαθμού 0. (Στο σχήμα οι κορυφές $v_1, v_2, \ldots, v_8$ αναπαρίστανται με την σειρά από τα αριστερά προς τα δεξιά και σημειώνονται μόνο οι τιμές της ακολουθίας a_i).

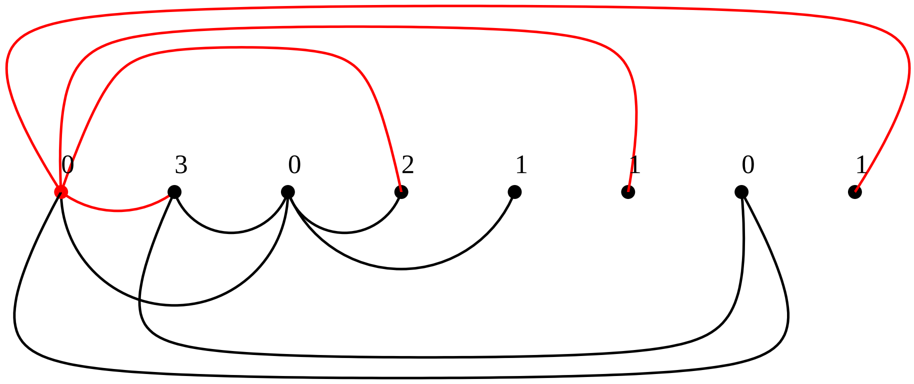
Επιλέγουμε (αυθαίρετα) την κορυφή v_3 (που είναι μαρκαρισμένη με κόκκινο). Η v_3 έχει $a_3 = 4$ οπότε την συνδέουμε με τις 4 κορυφές οι οποίες έχουν τις μεγαλύτερες δυνατές τιμές της ακολουθίας a_i, εδώ είναι οι κορυφές v_1, v_2, v_4 και v_5, οπότε προκύπτει το επόμενο γράφημα στο οποίο έχουμε ενημερώσει τις αντίστοιχες τιμές των a_i.

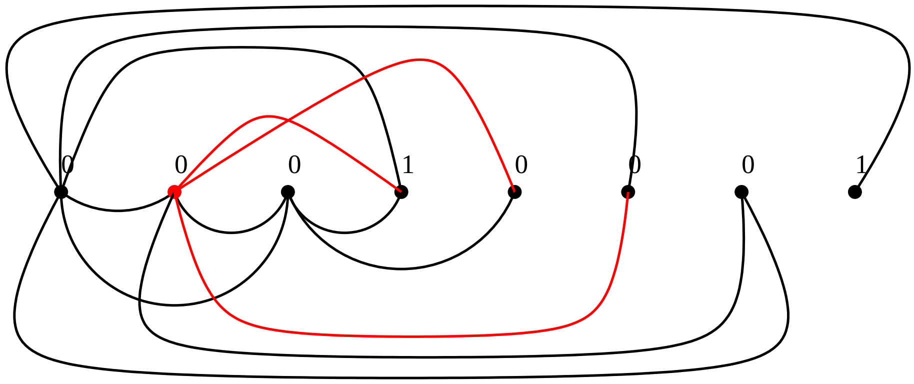


Στην συνέχεια, επιλέγουμε (αυθαίρετα) την κορυφή v_7 για την οποία $a_7 = 2$ και την συνδέουμε με τις 2 κορυφές οι οποίες έχουν τις μεγαλύτερες δυνατές τιμές της ακολουθίας a_i, εδώ είναι οι κορυφές v_1, v_2, οπότε προκύπτει το επόμενο γράφημα στο οποίο έχουμε ενημερώσει τις αντίστοιχες τιμές των a_i.

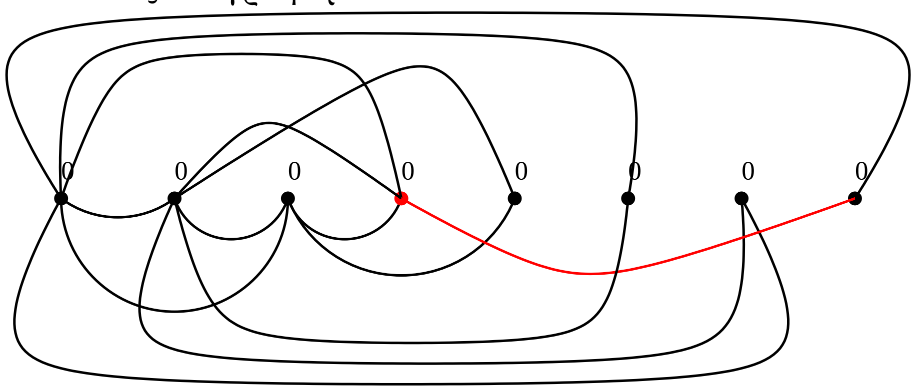


Συνεχίζουμε με τον ίδιο τρόπο, επιλέγοντας κορυφές v_k με $a_k > 0$, οπότε προκύπτουν διαδοχικά τα γραφήματα:





και τέλος το γράφημα

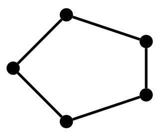


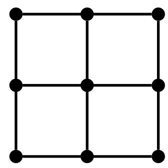
το οποίο έχει ακολουθία βαθμών $(6, 6, 4, 4, 2, 2, 2, 2)$.

Παρατηρήσεις : Στην περίπτωση όπου η ακολουθία $(d_1, d_2, \ldots, d_n)$ δεν είναι γραφική ο αλγόριθμος θα αποτύχει διότι θα υπάρχουν θετικά a_k αλλά δεν θα υπάρχουν αρκετές διαθέσιμες κορυφές για να συνδεθεί η κορυφή v_k.

Ασκήσεις προς επίλυση

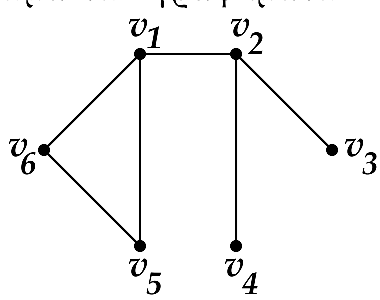
(1) Χρησιμοποιώντας την βιβλιοθήκη networkx κατασκευάστε τα παρακάτω γραφήματα

C_5 

$L_{3,3}$ 

(2) Ένα γράφημα G έχει 20 κορυφές.
  i) Ποιος είναι ο ελάχιστος και ο μέγιστος δυνατός βαθμός των κορυφών του;
  ii) Ποιος είναι ο ελάχιστος και ο μέγιστος δυνατός αριθμός των δεσμών που περιέχει;
  iii) Αν το G έχει 50 δεσμούς, πόσους δεσμούς έχει το συμπλήρωμα του;

(3) Να βρεθεί το συμπλήρωμα των γραφημάτων

G 

$H = K_6$

(4) Να κατασκευασθεί
  i) ένα 2-κανονικό γράφημα με 10 κορυφές.
  ii) ένα 3-κανονικό γράφημα με 10 κορυφές.

(5) Να εξετασθεί, αν υπάρχουν, γραφήματα δεσμών G που έχουν τις παρακάτω ακολουθίες βαθμών.
  i) $(11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0)$
  ii) $(5, 5, 4, 4, 3, 3, 2, 2, 1, 1)$.
  iii) $(5, 5, 3, 3, 3, 1)$
  iv) $(6, 3, 3, 3, 3, 3, 3)$
  v) $(3, 3, 3, 3, 1, 1, 1, 1, 1, 1)$
  vi) $(3, 3, 3, 3, 2, 2, 2, 2, 2)$.
  vii) $(4, 4, 2, 2, 2, 2, 2, 2, 2, 2)$.

(6) Να δειχθεί ότι για κάθε $n \geq 2$ υπάρχει γράφημα δεσμών G με $2n$ κορυφές και ακολουθία βαθμών $(n, n, n-1, n-1, \ldots, 3, 3, 2, 2, 1, 1)$.

2. ΣΥΝΕΚΤΙΚΟΤΗΤΑ

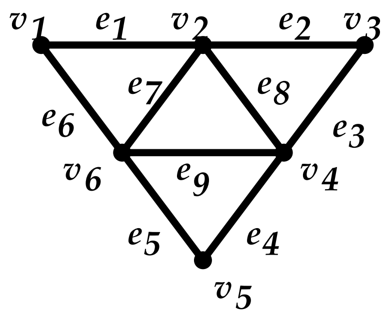
Διαδρομή (walk) που ενώνει τους κόμβους v_i, v_j ενός γράφηματος G (ή $v_i - v_j$ **διαδρομή**) είναι μια ακολουθία της μορφής $(v_i, e_{ik}, v_k, e_{kl}, v_l, \ldots, v_r, e_{rj}, v_j)$, όπου e_{st} είναι ο δεσμός του γραφήματος που ενώνει τους κόμβους v_s και v_t. (Συνήθως περιγράφουμε μια διαδρομή μόνο με τους διαδοχικούς κόμβους της: $(v_i, v_k, v_l, \ldots, v_r, v_j)$).

Μήκος (length) μιας διαδρομής ονομάζεται το πλήθος των δεσμών της.

Αν σε μια $v_i - v_j$ διαδρομή του G κάθε δεσμός εμφανίζεται μια μόνο φορά, η διαδρομή λέγεται $v_i - v_j$ **δρόμος** (trail) του G. Αν επιπλέον, σε ένα $v_i - v_j$ δρόμο του G κάθε κόμβος εμφανίζεται μια μόνο φορά, ο δρόμος λέγεται $v_i - v_j$ **μονοπάτι** (path) του G.

Μια $v_i - v_j$ διαδρομή ή ένας $v_i - v_j$ δρόμος του G, με $v_i = v_j$ λέγεται **κλειστή διαδρομή** του G ή **κλειστός δρόμος** του G. Τέλος, ένας κλειστός δρόμος του G μήκους n, με n κορυφές λέγεται **κύκλος** (cycle) του G.

Παράδειγμα: Για το γράφημα G έχουμε:



$v_1 - v_5$ διαδρομή του G: $(v_1, e_1, v_2, e_7, v_6, e_9, v_4, e_8, v_2, e_7, v_6, e_5, v_5)$, ή συντομότερα $(v_1, v_2, v_6, v_4, v_2, v_6, v_5)$.

$v_1 - v_5$ δρόμος του G: $(v_1, v_2, v_3, v_4, v_2, v_6, v_5)$.

$v_1 - v_5$ μονοπάτι του G: $(v_1, v_2, v_3, v_4, v_6, v_5)$.

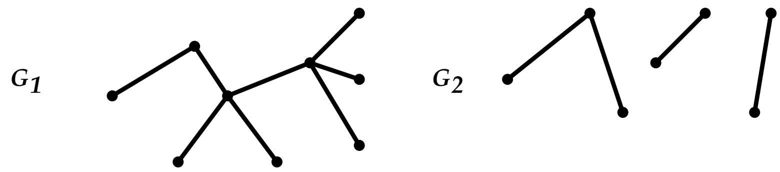
Κλειστή διαδρομή του G: $(v_1, v_2, v_6, v_4, v_3, v_2, v_6, v_1)$.

Κλειστός δρόμος του G: $(v_1, v_2, v_3, v_4, v_2, v_6, v_1)$.

Κύκλος του G: $(v_1, v_2, v_4, v_6, v_1)$.

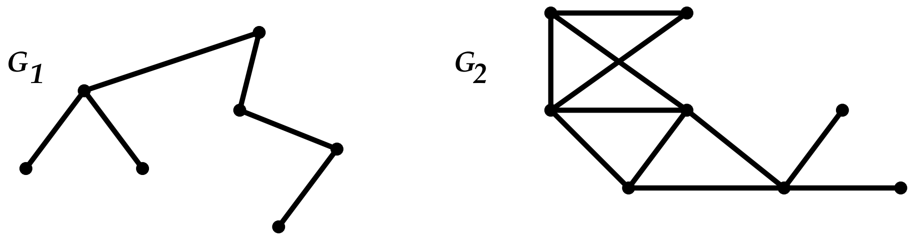
Άκυκλο λέγεται ένα γράφημα που δεν έχει κύκλους.

Παραδείγματα:

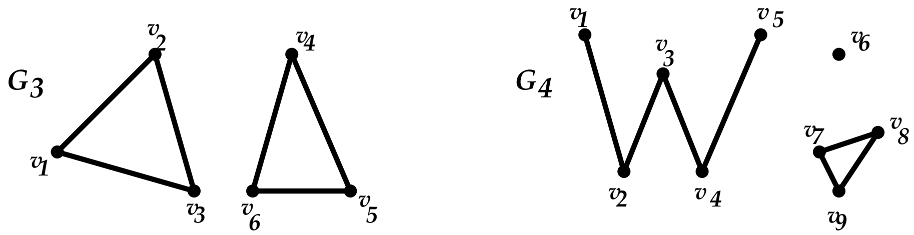


Ένα γράφημα λέγεται **συνεκτικό** αν για οποιουσδήποτε δύο κόμβους του, υπάρχει μονοπάτι που τους ενώνει.

Παραδείγματα:



Συνεκτικά γραφήματα



Μη συνεκτικά γραφήματα

Συνιστώσα (connected component) ενός γραφήματος G ονομάζεται κάθε μεγιστικό (maximal) συνεκτικό υπογράφημά του (δηλαδή κάθε συνεκτικό υπογράφημά του που δεν είναι υπογράφημα κάποιου άλλου συνεκτικού υπογραφήματος του G).

Προφανώς τα συνεκτικά γραφήματα αποτελούνται από μια μόνο συνιστώσα: τον εαυτό τους.

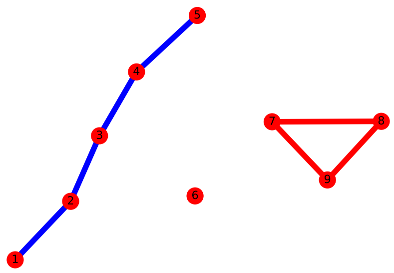
Παραδείγματα: Στο προηγούμενο σχήμα οι συνιστώσες του G_3 είναι τα δύο τρίγωνα $G_{3,1}$, $G_{3,2}$ με $V(G_{3,1}) = \{v_1, v_2, v_3\}$ και $V(G_{3,2}) = \{v_4, v_5, v_6\}$ αντίστοιχα, ενώ το G_4 έχει προφανώς τρεις συνιστώσες.

Μπορούμε να βρούμε τις συνεκτικές συνιστώσες ενός γραφήματος με την βοήθεια της μεθόδου `connected_components(G)`. Στο επόμενο πρόγραμμα σχεδιάζουμε με χρώματα τους δεσμούς τους ανάλογα με το μέγεθος κάθε συνιστώσας.

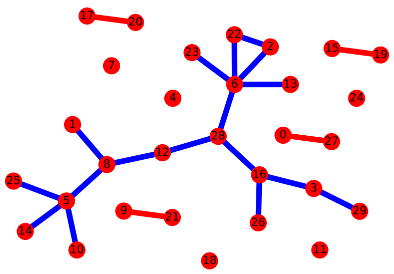
```
import networkx as nx
import matplotlib.pyplot as plt

G = nx.Graph()
V = [v for v in range(1,10)]
E = [[1,2],[2,3],[3,4],[4,5],[7,8],[7,9],[8,9]]
G.add_nodes_from(V)
G.add_edges_from(E)

#draw the graph using graphviz layout positioning algorithm
#keep the positions of nodes in order to redraw
pos = nx.drawing.nx_agraph.graphviz_layout(G)
nx.draw(G,pos,with_labels=True)

#find the connected components of G
#sort the list from the largest to smaller
```

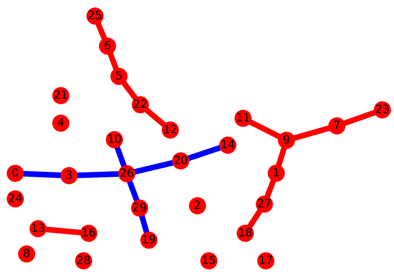
```
Gcc = sorted(nx.connected_components(G), key=len, reverse=True)

#G0 the largest connected components
G0 = G.subgraph(Gcc[0])
#Draw the edges of the largest component
nx.draw_networkx_edges(G0,pos,edge_color='b',width=6.0)

#for every connected components of size > 1 draw its edges
for cc in Gcc[1:]:
    if len(cc) > 1:
        G1 = G.subgraph(cc)
        nx.draw_networkx_edges(G1,pos,edge_color='r', width=6.0)

plt.savefig("lect02a.eps")
plt.show()
```

Output:



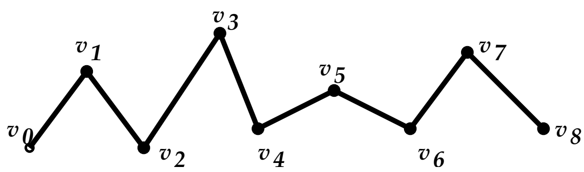
Δύο επιπλέον παραδείγματα όπου το γράφημα G έχει κατασκευασθεί με την μέθοδο `nx.gnp_random_graph(30,0.054)`





Ένα συνεκτικό γράφημα $G = (V, E)$ με $V = \{v_0, v_1, \ldots, v_n\}$ λέγεται $v_0 - v_n$ **μονοπάτι** (ή απλά **μονοπάτι**) μήκους n, αν $d(v_0) = d(v_n) = 1$ και $d(v_i) = 2$, για κάθε $i \neq 0, n$.

Παράδειγμα:

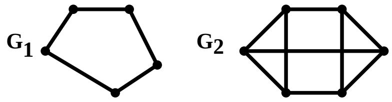


Μονοπάτι μήκους 8

Ένα συνεκτικό 2-κανονικό γράφημα λέγεται **κύκλος**, ενώ ένα 3-κανονικό γράφημα λέγεται **κυβικό γράφημα**.

Ο κύκλος με n κορυφές συμβολίζεται με C_n.

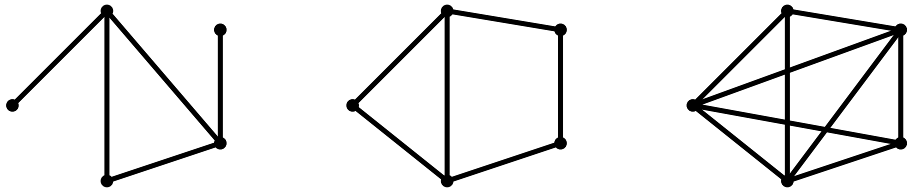
Παραδείγματα:



Τα γραφήματα G_1, G_2 είναι ο κύκλος C_5 και ένα κυβικό γράφημα αντίστοιχα.

Παρατήρηση: Παρατηρήστε τη διαφορά ανάμεσα στους ορισμούς «μονοπάτι γραφήματος» και «κύκλος γραφήματος» που δόθηκαν νωρίτερα και στους ορισμούς των γραφημάτων «μονοπάτι» και «κύκλος» που δίνονται εδώ.

Ορισμένα συνεκτικά γραφήματα είναι "πιο συνδεδεμένα" από ότι άλλα.



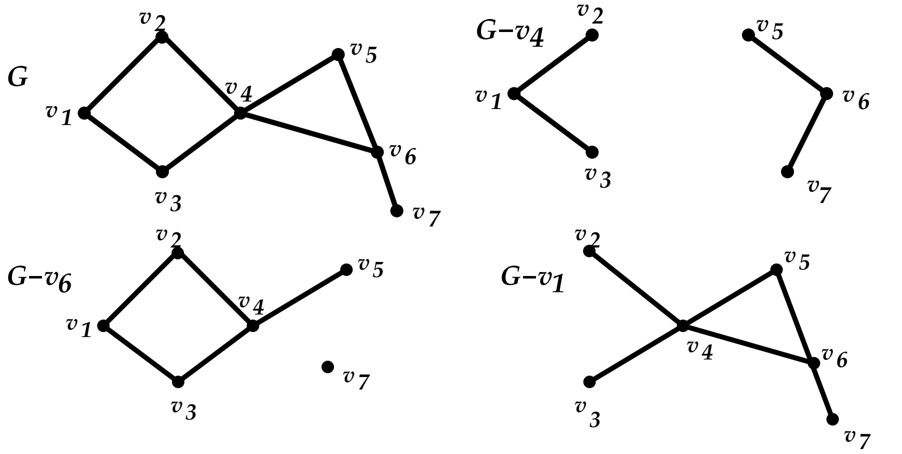
Για παράδειγμα, κάποια συνεκτικά γραφήματα έχουν κορυφές ή δεσμούς που αν τους αφαιρέσουμε, τότε το γράφημα παύει να είναι συνεκτικό, ενώ άλλα γραφήματα παραμένουν συνεκτικά ακόμα και όταν αφαιρέσουμε μια ή περισσότερες κορυφές ή δεσμούς τους.

Ο υπολογισμός του αριθμού των κορυφών ή δεσμών που πρέπει να αφαιρεθούν για να προκύψει μη συνεκτικό γράφημα είναι ένα μέτρο της ανθεκτικότητας σε αποτυχίες ενός δικτύου επικοινωνίας.

Διαισθητικά, η ανθεκτικότητα ενός γραφήματος εξαρτάται από τον αριθμό των εναλλακτικών μονοπατιών μεταξύ των ζευγών κορυφών του. Όσο μεγαλύτερος είναι αυτός ο αριθμός τόσο πιο ανθεκτικό είναι το γράφημα σε αποτυχίες κορυφών ή δεσμών.

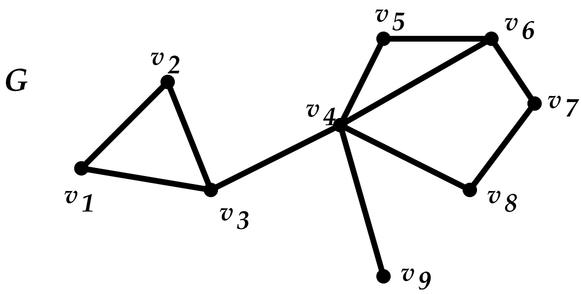
Κλείδωση (articulation point), ή **σημείο κοπής** (cut point) ενός συνεκτικού γραφήματος G λέγεται κάθε $v \in V$ τέτοιο ώστε $G - v$: μη συνεκτικό.

Παραδείγματα: Οι κόμβοι v_4, v_6 του παρακάτω γραφήματος G είναι κλειδώσεις, ενώ ο v_1 δεν είναι.

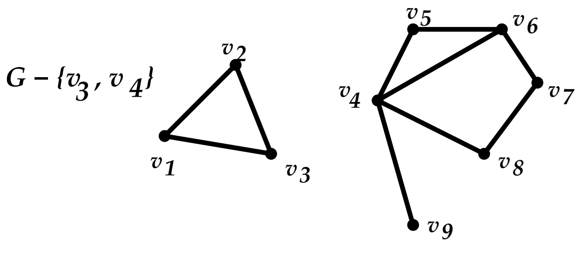


Προφανώς, κλειδώσεις ενός μη συνεκτικού γραφήματος G ονομάζονται οι κλειδώσεις των συνιστωσών του G.

Γέφυρα (bridge) ή **ισθμός** ενός συνεκτικού γραφήματος G λέγεται κάθε $e \in E$ τέτοιος ώστε $G - e$: μη συνεκτικό.
Παράδειγμα:
Για το γράφημα



ο $\{v_3, v_4\}$ είναι γέφυρα, αφού το γράφημα



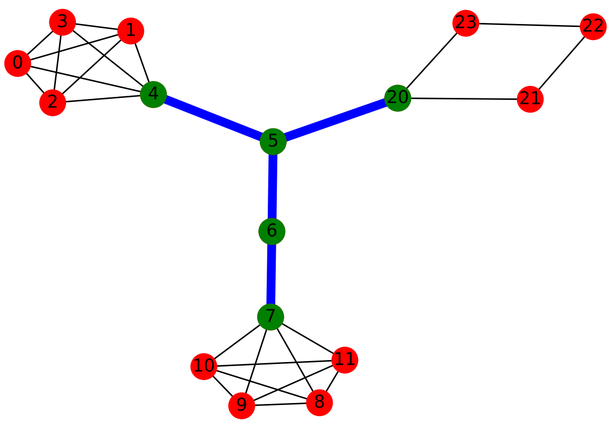
είναι μη συνεκτικό.
Προφανώς, γέφυρες ενός μη συνεκτικού γραφήματος G ονομάζονται οι γέφυρες των συνιστωσών του G.

Μπορούμε να εντοπίσουμε τις γέφυρες και τις κλειδώσεις ενός γραφήματος χρησιμοποιώντας τις μεθόδους bridges(G) και articulation_points(G).

```
import networkx as nx
import matplotlib.pyplot as plt

G = nx.barbell_graph(5,2)
G.add_edges_from([[5,20],[20,21],[21,22],[22,23],[23,20]])

pos = nx.drawing.nx_agraph.graphviz_layout(G)
nx.draw(G,pos,with_labels=True)
#nx.draw_networkx(G)
B = list(nx.bridges(G))
print("Bridges of G:",B)
for e in B:
    G1 = G.subgraph(e)
    nx.draw_networkx_edges(G1,pos,edge_color='blue',width=6.0)
C = list(nx.articulation_points(G))
print("Cut points of G:",C)
for v in C:
    G1 = G.subgraph(v)
    nx.draw_networkx_nodes(G1,pos,node_color='green',width=6.0)
plt.savefig("barbell52.eps")
plt.show()
```

Output:

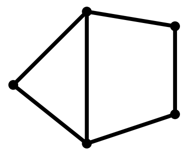
```
Bridges of G: [(4, 5), (5, 6), (5, 20), (6, 7)]
Cut points of G: [7, 6, 5, 20, 4]
```



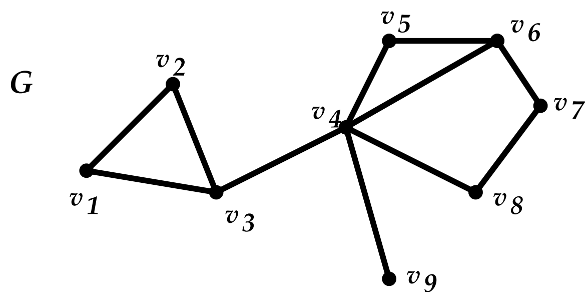
Πρόταση 2. *Έστω $G = (V, E)$ ένα συνεκτικό γράφημα και $v \in V$. Οι παρακάτω προτάσεις είναι ισοδύναμες:*
i) Ο κόμβος v είναι κλείδωση του G.
ii) Υπάρχει μια διαμέριση του $V \setminus \{v\}$ σε υποσύνολα U, W τέτοια ώστε, για κάθε $u \in U$ και για κάθε $w \in W$ ο κόμβος v ανήκει σε κάθε $u - w$ μονοπάτι.
iii) Υπάρχουν κόμβοι u, w διάφοροι του v τέτοιοι ώστε ο v ανήκει σε κάθε $u - w$ μονοπάτι.

Πρόταση 3. *Έστω $G = (V, E)$ ένα συνεκτικό γράφημα και $e \in E$. Οι παρακάτω προτάσεις είναι ισοδύναμες:*
i) Ο δεσμός e είναι ισθμός.
ii) Ο δεσμός e δεν ανήκει σε κανένα κύκλο του G.
iii) Υπάρχει διαμέριση του V σε U, W τέτοια ώστε για κάθε $u \in U$, $w \in W$ ο δεσμός e ανήκει σε κάθε $u - w$ μονοπάτι.
iv) Υπάρχουν $u, w \in V$ τέτοιοι ώστε ο δεσμός e ανήκει σε κάθε $u - w$ μονοπάτι.

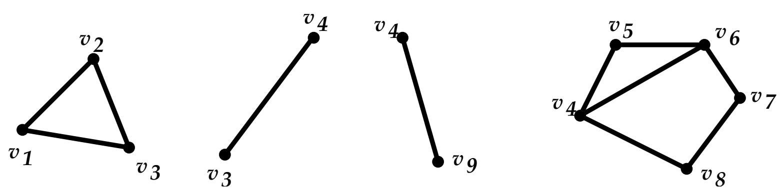
Ένα μη τετριμμένο, συνεκτικό γράφημα χωρίς κλειδώσεις λέγεται **μη διαχωρίσιμο** (ή **συμπαγές**, ή **δισυνεκτικό**).
Παράδειγμα: Το παρακάτω γράφημα είναι μη διαχωρίσιμο:



Αν το H είναι ένα μεγιστικό μη διαχωρίσιμο υπογράφημα του G (δηλαδή το H δεν είναι υπογράφημα κάποιου άλλου μη διαχωρίσιμου υπογραφήματος του G) τότε λέγεται **μπλοκ** (ή **δισυνεκτική συνιστώσα**) του G.
Παράδειγμα: Τα μπλοκ του γραφήματος



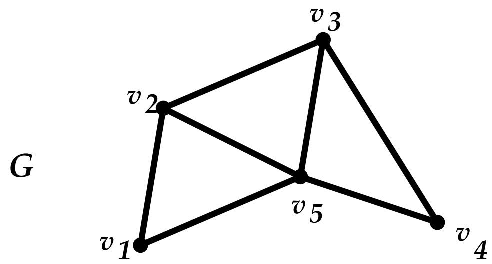
είναι τα



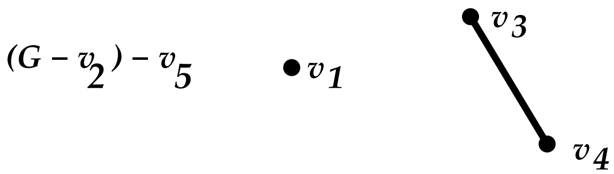
Σύνολο κλειδώσεων ενός συνεκτικού γραφήματος G λέγεται κάθε $\{v_1, v_2, \ldots, v_n\} \subseteq V$ τέτοιο ώστε το $(((G - v_1) - v_2) - \cdots) - v_n$ να είναι μη συνεκτικό.

Ένα γράφημα ονομάζεται k-**συνεκτικό** (k-connected) αν κάθε σύνολο κλειδώσεων του περιέχει τουλάχιστον k κορυφές. Με άλλα λόγια, αν το G είναι k-συνεκτικό τότε το γράφημα που προκύπτει από την διαγραφή οποιουδήποτε συνόλου $k - 1$ κορυφών του G είναι επίσης συνεκτικό.

Παράδειγμα: Το γράφημα



είναι 2-συνεκτικό, αφού το σύνολο $\{v_2, v_5\}$ είναι ένα ελάχιστο σύνολο κλειδώσεων:



Παρατήρηση. Αν ένα γράφημα G είναι (k+1)-συνεκτικό, τότε είναι και k-συνεκτικό. Πράγματι, αφού το G είναι $(k+1)$-συνεκτικό η διαγραφή οποιονδήποτε k κορυφών του, δεν το κάνει μη συνεκτικό, άρα ούτε και η διαγραφή $k-1$ κορυφών οδηγεί σε μη συνεκτικό γράφημα, οπότε το G είναι και k-συνεκτικό. Δεν ισχύει το αντίστροφο, αν ένα γράφημα είναι k-συνεκτικό τότε δεν είναι $(k + 1)$-συνεκτικό.

Στην επόμενη πρόταση δίδεται μια απλή αναγκαία και ικανή συνθήκη ώστε ένα γράφημα να είναι 2-συνεκτικό.

Πρόταση 4. *Ένα γράφημα είναι 2-συνεκτικό αν και μόνο αν για κάθε ζεύγος κορυφών του υπάρχει τουλάχιστον ένα κύκλος που τις περιέχει.*

Παράδειγμα : Στο προηγούμενο 2-συνεκτικό γράφημα οι κορυφές v_2 και v_4 ανήκουν στον κύκλο $(v_2v_5v_4v_3v_2)$, οι κορυφές v_2 και v_1 ανήκουν στον κύκλο (v_1, v_2, v_5, v_1), οι κορυφές v_2 και v_3 ανήκουν στον κύκλο $(v_2, v_1, v_5, v_4, v_3, v_3)$, κ.ο.κ.

Παρατήρηση. Αν ένα γράφημα είναι 2-συνεκτικό τότε για κάθε ζεύγος κορυφών του υπάρχουν 2 τουλάχιστον διαφορετικά μονοπάτια που τους συνδέουν τα οποία, εκτός από τα άκρα τους, περιέχουν διαφορετικές κορυφές το καθένα.

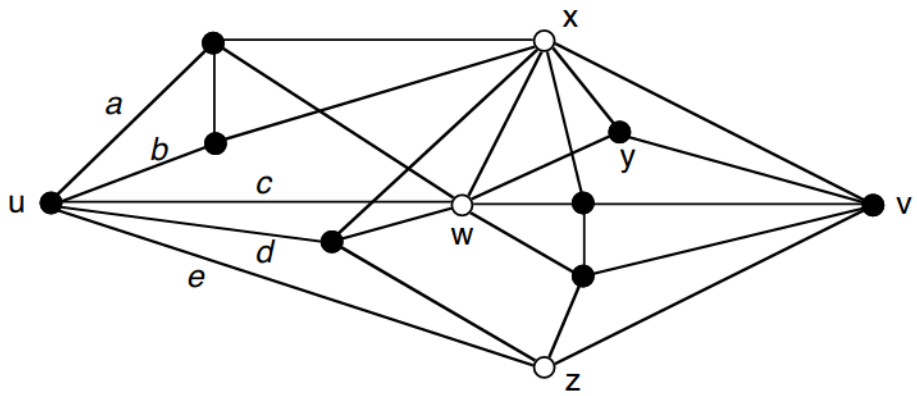
Γενικότερα έχει αποδειχθεί ότι ένα γράφημα G είναι k-συνεκτικό αν και μόνο αν κάθε ζεύγος κορυφών του u, v υπάρχουν k τουλάχιστον διαφορετικά μονοπάτια που τους συνδέουν τα οποία, εκτός από τα άκρα τους, περιέχουν διαφορετικές κορυφές το καθένα. (Η απόδειξη δίδεται αργότερα.)

3. ΘΕΩΡΗΜΑ MENGER

Δύο ή περισσότερα μονοπάτια ενός γραφήματος ονομάζονται **ανεξάρτητα** αν κανένα από αυτά δεν περιέχει ως εσωτερική κορυφή κάποια κορυφή του άλλου. Ειδικότερα, δύο $u - v$ μονοπάτια είναι ανεξάρτητα αν και μόνο αν τα άκρα τους u, v είναι οι μοναδικές κοινές κορυφές τους.

Έστω u, v δύο μη γειτονικές κορυφές και $X \subseteq V \setminus \{u, v\}$ τέτοιο ώστε κάθε $u - v$ μονοπάτι να περιέχει μια κορυφή του X, τότε επειδή οι κορυφές u, v περιέχονται σε διαφορετικές συνιστώσες του γραφήματος $G - X$ λέμε ότι το X **διαχωρίζει** (separates) τις κορυφές u, v και ονομάζεται (u, v)-**διαχωριστής**. Επιπλέον, ένα σύνολο κορυφών X ονομάζεται **διαχωριστής** του G αν διαχωρίζει κάποιο ζεύγος κορυφών του G.

Παράδειγμα: Οι κορυφές $\{x, z, w\}$ αποτελούν έναν (u, v)-διαχωριστή του επόμενου γραφήματος, διότι κάθε u-v μονοπάτι περιέχει τουλάχιστον μια από αυτές τις κορυφές.



Με βάση τα προηγούμενα έχουμε δύο διαφορετικές οπτικές της συνεκτικότητας ενός γραφήματος.

Η μια είναι ο αριθμός των κορυφών ή δεσμών που απαιτούνται να αφαιρεθούν ώστε να προκύψει μη συνεκτικό γράφημα (ή γενικότερα να αυξηθεί ο αριθμός των συνιστωσών του γραφήματος) και από την άλλη είναι ο αριθμός των διαφορετικών ανεξάρτητων μονοπατιών που ενώνουν οποιοδήποτε ζεύγος κορυφών.

Σε αυτές τις δύο οπτικές αντιστοιχούν τα επόμενα προβλήματα βελτιστοποίησης για δύο μη γειτονικές κορυφές u, v ενός συνεκτικού γραφήματος G.

- **Πρόβλημα μεγιστοποίησης** Να προσδιορισθεί ο μέγιστος αριθμός ανεξάρτητων $u - v$ μονοπατιών.
- **Πρόβλημα ελαχιστοποίησης** Να προσδιορισθεί ο ελάχιστος αριθμός των κορυφών που απαιτούνται για να διαχωρίσουν τις κορυφές u, v.

Παρατηρήστε ότι

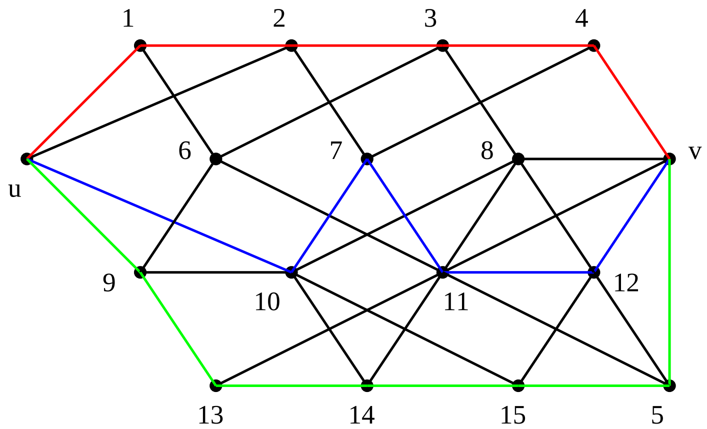
ο μέγιστος αριθμός ανεξάρτητων $u - v$ **μονοπατιών είναι μικρότερος ή ίσος από τον ελάχιστο αριθμό κορυφών που απαιτούνται για να διαχωρίσουν τις κορυφές** u, v.

Πράγματι, αν το γράφημα περιέχει k ανεξάρτητα $u - v$ μονοπάτια απαιτούνται τουλάχιστον k κορυφές για τον διαχωριστή (πρέπει να επιλέξουμε τουλάχιστον μια κορυφή από κάθε μονοπάτι). Αντίστοιχα, αν το γράφημα περιέχει ένα διαχωριστή μεγέθους k τότε αφού κάθε μονοπάτι διέρχεται από το πολύ μια από τις κορυφές του διαχωριστή μπορούν να υπάρχουν το πολύ k ανεξάρτητα μονοπάτια.

Το βασικό αποτέλεσμα αυτής της ενότητας είναι το θεώρημα του Menger το οποίο δείχνει ότι οι δύο αριθμοί είναι ίσοι.

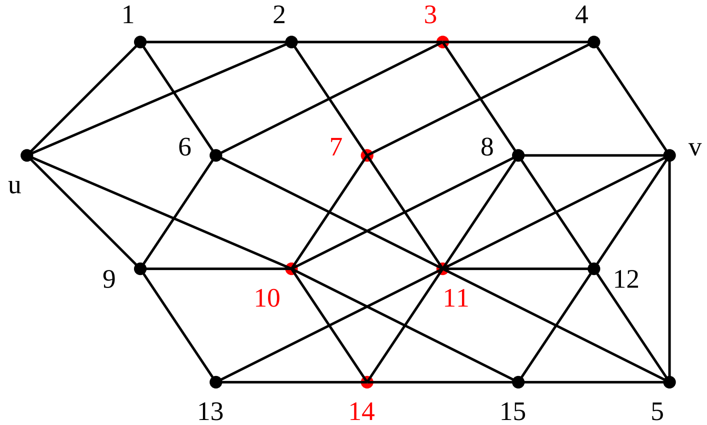
Παρατήρηση: Το θεώρημα του Menger και οι παραλλαγές του αποτελούν την επιτομή της σχέσης **πρωτεύοντος-δυϊκού** (primal-dual). Ο όρος πρωτεύον - δυικό προέρχεται από την επιχειρησιακή έρευνα όπου εμφανίζονται ζεύγη προβλημάτων βελτιστοποίησης τα οποία είναι άρρηκτα συνδεδεμένα μεταξύ τους. Συνήθως, το ένα από αυτά είναι ένα πρόβλημα μεγιστοποίησης μιας αντικειμενικής (objective) συνάρτησης, ενώ το άλλο είναι ένα πρόβλημα ελαχιστοποίησης μιας άλλης αντικειμενικής συνάρτησης. Μια εφικτή λύση για το ένα από τα δύο προβλήματα δίνει ένα φράγμα για την βέλτιστη λύση στο άλλο πρόβλημα (ασθενής δυϊκότητα) και η βέλτιστη τιμή του ενός προβλήματος ισούται με την βέλτιστη τιμή του άλλου (ισχυρή δυϊκότητα).

Παράδειγμα: Η ύπαρξη των επόμενων 3 ανεξάρτητων μονοπατιών ($(u, 1, 2, 3, 4, v)$, $(u, 10, 7, 11, 12, v)$, $(u, 9, 13, 14, 15, 5, v)$) στο παρακάτω γράφημα



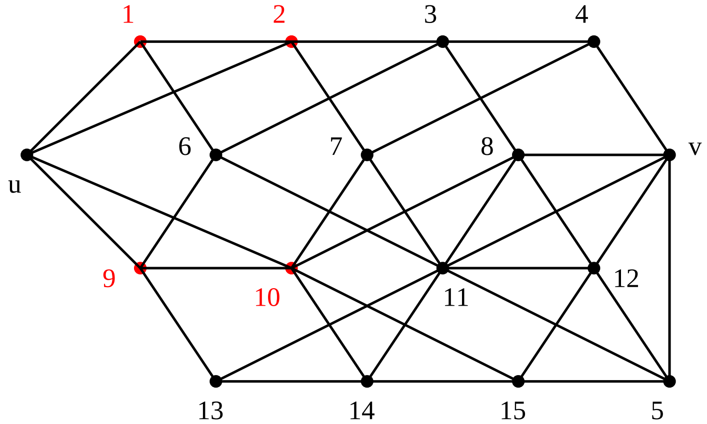
έχει ως συνέπεια ότι ο ελάχιστος αριθμός κορυφών που απαιτούνται για να διαχωρίσουν τις κορυφές u και v είναι τουλάχιστον 3 (αφού πρέπει να επιλέξουμε τουλάχιστον μια κορυφή από κάθε ανεξάρτητο μονοπάτι).

Ομοίως, η ύπαρξη ενός διαχωριστή με 5 κορυφές (τις 3, 7, 10, 11, 14)



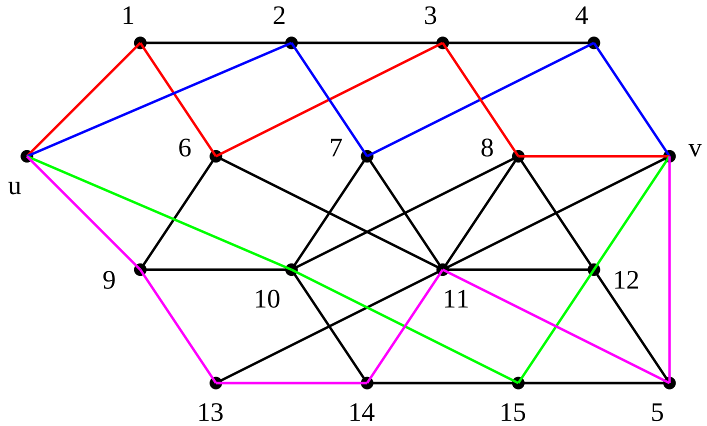
έχει ως συνέπεια ότι ο μέγιστος αριθμός ανεξάρτητων $u-v$ μονοπατιών είναι το πολύ 5 (αφού κάθε μονοπάτι πρέπει να διέρχεται από το πολύ μια από τις κορυφές του διαχωριστή).

Όμως, η εύρεση ενός διαχωριστή με 4 κορυφές (τις 1, 2, 9, 10)



έχει ως συνέπεια ότι ο μέγιστος αριθμός ανεξάρτητων $u-v$ μονοπατιών είναι το πολύ 4.

Τέλος, η εύρεση των επόμενων 4 ανεξάρτητων μονοπατιών ($(u,1,6,3,8,v)$, $(u,2,7,4,v)$, $(u,10,15,12,v)$, $(u,9,13,14,11,5,v)$)

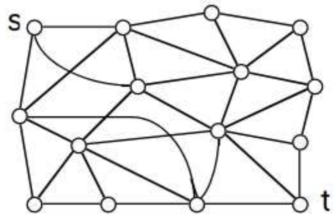


έχει ως συνέπεια ότι ο ελάχιστος αριθμός κορυφών που απαιτούνται για να διαχωρίσουν τις κορυφές u και v είναι τουλάχιστον 4.

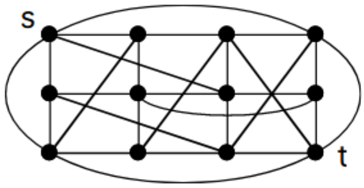
Άρα, επειδή ο μέγιστος αριθμός ανεξάρτητων $u-v$ μονοπατιών είναι μικρότερος ή ίσος από τον ελάχιστο αριθμό κορυφών κάθε (u,v)-διαχωριστή έπεται ότι ο ζητούμενος αριθμός είναι και για τα δύο προβλήματα ίσος με 4.

Πρόταση 5 (**Θεώρημα Menger, 1927**). *Έστω u, v δύο κορυφές του G. Αν $\{u, v\} \notin E$ τότε ο ελάχιστος αριθμός κορυφών που διαχωρίζουν την u από την v ισούται με τον μέγιστο αριθμό ανεξάρτητων $u - v$ μονοπατιών.*

Άσκηση 1. *Να δειχθεί ότι δεν υπάρχει διαχωριστής μεγέθους 2 για τις κορυφές s και t του επόμενου γραφήματος:*



Άσκηση 2. *Να υπολογισθεί ο ελάχιστος αριθμός κορυφών που διαχωρίζουν τις κορυφές s και t του επόμενου γραφήματος:*



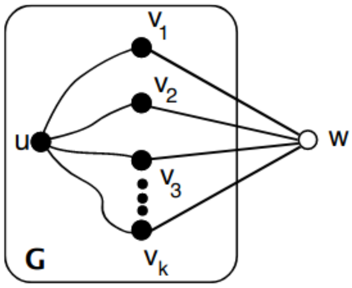
Πόρισμα 6 (Whitney, 1932). *Ένα γράφημα G είναι k-συνεκτικό αν και μόνο αν κάθε ζεύγος κορυφών του u, v ενώνεται με τουλάχιστον k ανεξάρτητα $u - v$ μονοπάτια.*

Απόδειξη. Αν ένα γράφημα G περιέχει k ανεξάρτητα μονοπάτια ανάμεσα σε οποιοδήποτε ζεύγος κορυφών του τότε δεν μπορεί να διαχωρισθεί με λιγότερες από k κορυφές, επομένως είναι k-συνεκτικό.

Αντίστροφα, έστω ότι το G είναι k-συνεκτικό (και έχει περισσότερες από k κορυφές) αλλά περιέχει κορυφές u, v που δεν ενώνονται με k ανεξάρτητα $u - v$ μονοπάτια. Τότε από το Θεώρημα του Menger (Πρόταση 5) οι u, v είναι γειτονικές. Έστω $G' = G - \{u, v\}$. Τότε το G' περιέχει το πολύ $k-2$ ανεξάρτητα $u-v$ μονοπάτια, άρα από το Θεώρημα του Menger (Πρόταση 5) μπορούμε να διαχωρίσουμε τις u, v στο G' με ένα σύνολο X που περιέχει το πολύ $k-2$ κορυφές. Επειδή το G περιέχει περισσότερες από k κορυφές υπάρχει κορυφή $w \notin X \cup \{u, v\}$ στο G. Αφού το X είναι διαχωριστής των u και v, θα X διαχωρίζει την w από τουλάχιστον μια από τις u, v, και ας υποθέσουμε ότι διαχωρίζει τις w και u. Τότε το σύνολο $X \cup \{v\}$ περιέχει το πολύ $k - 1$ κορυφές και διαχωρίζει τις w και u στο G, το οποίο είναι άτοπο, αφού το G είναι k-συνεκτικό. □

Πόρισμα 7. *Έστω G ένα k-συνεκτικό γράφημα. Για κάθε υποσύνολο u, v_1, v_2, …, v_k που περιέχει $k + 1$ κορυφές του G υπάρχουν μονοπάτια $u - v_1$, $u - v_2$, …, $u - v_k$ τα οποία είναι ανά δύο ανεξάρτητα.*

Απόδειξη. Από το G κατασκευάζουμε ένα νέο γράφημα $G \cup \{w\}$ προσθέτοντας μια νέα κορυφή w η οποία ενώνεται με δεσμούς με τις κορυφές v_1, v_2, …, v_k.

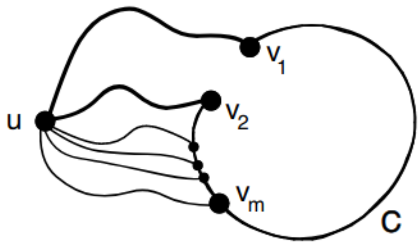


Αφού το G είναι k-συνεκτικό έπεται ότι και το $G \cup \{w\}$ είναι επίσης k-συνεκτικό. (Άσκηση).

Επομένως, υπάρχουν k ανεξάρτητα μονοπάτια $u - w$. Κάθε τέτοιο μονοπάτι υποχρεωτικά θα έχει ως άκρο ακριβώς ένα από τους δεσμούς $\{w, v_i\}$, $i \in [k]$. Άρα, τα μονοπάτια $u - v_i$ που προκύπτουν από αυτά σβήνοντας τον τελευταίο δεσμό τους είναι ανεξάρτητα. □

Πόρισμα 8 (Dirac, 1960). *Έστω G ένα k-συνεκτικό γράφημα με τουλάχιστον $k+1$ κορυφές, όπου $k \geq 3$, και έστω U ένα σύνολο k κορυφών του G. Τότε υπάρχει κύκλος του G που περιέχει όλες τις κορυφές του U.*

Απόδειξη. Έστω C ένας κύκλος του G που περιέχει τον μέγιστο δυνατό αριθμό κορυφών του U και έστω ότι $\{v_1, v_2, \ldots, v_m\}$, $m \leq k$ είναι οι κορυφές του U που περιέχονται στον C. Επειδή το G είναι τουλάχιστον 2-συνεκτικό έπεται ότι $m \geq 2$. Έστω ότι υπάρχει κορυφή $u \in U$ η οποία δεν περιέχεται στον κύκλο C. Τότε υπάρχουν ανεξάρτητα μονοπάτια $u - v_1$ και $u - v_2$.



Επομένως, μπορούμε να επεκτείνουμε τον κύκλο αφαιρώντας τον δεσμό $\{v_1, v_2\}$ και προσθέτοντας τα μονοπάτια $v_1 - u$ και $u - v_2$, κατασκευάζοντας ένα κύκλο με περισσότερες κορυφές του U, το οποίο είναι άτοπο. □

Υπάρχουν πολλές παραλλαγές του θεωρήματος Menger. Για παράδειγμα, αντί δύο κορυφές μπορεί να έχουμε δύο σύνολα κορυφών. Έστω $A, B \subseteq V$. Ένα μονοπάτι $P = (v_1, v_2, \ldots, v_k)$ με $v_1 \in A$ και $v_k \in B$ ονομάζεται $A - B$ **μονοπάτι**. Αν $A, B \subseteq V$ και $X \subseteq V$ τέτοιο ώστε κάθε $A - B$ μονοπάτι να περιέχει μια κορυφή του X τότε λέμε ότι το X **διαχωρίζει** (separates) τα σύνολα A και B και ονομάζεται (A, B)-**διαχωριστής**.

Πρόταση 9 (Γενίκευση του Θεωρήματος Menger). *Έστω $G = (V, E)$ ένα γράφημα δεσμών και $A, B \subseteq V$. Ο ελάχιστος αριθμός των κορυφών που διαχωρίζουν το A από το B ισούται με τον μέγιστο αριθμό των ανεξάρτητων $A - B$ μονοπατιών.*

3.1. Κατασκευή k-συνεκτικών γραφημάτων.

Ένα δίκτυο επικοινωνίας ονομάζεται **ανθεκτικό στα σφάλματα** (fault-tolerant) αν και μόνο αν υπάρχουν τουλάχιστον δύο διαφορετικά μονοπάτια ανάμεσα σε κάθε ζεύγος κορυφών. Η έννοια αυτή ταυτίζεται με την έννοια των 2-συνεκτικών γραφημάτων.

Έστω H ένα υπογράφημα του G. Ένα (μη τετριμμένο) μονοπάτι P του G ονομάζεται **H-μονοπάτι** αν και μόνο αν οι μόνες κοινές κορυφές του P με το H είναι τα άκρα του μονοπατιού. Το υπογράφημα του G που αποτελείται από το H και το H-μονοπάτι λέμε ότι προκύπτει από το H με **προσθήκη μονοπατιού**.

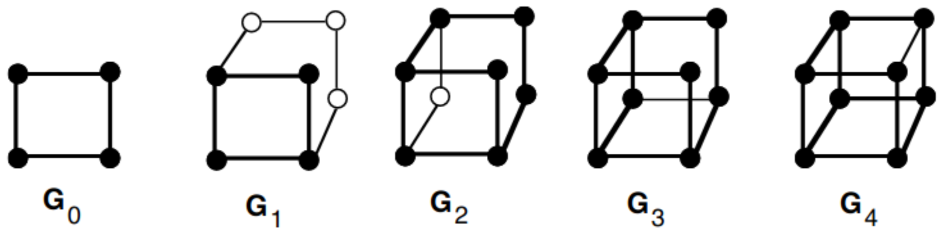
Λήμμα 10. *Έστω H ένα 2-συνεκτικό γράφημα. Τότε το γράφημα G που προκύπτει από την προσθήκη ενός H-μονοπατιού στο H είναι επίσης 2-συνεκτικό.*

Απόδειξη. Εύκολα προκύπτει ότι με την προσθήκη του H-μονοπατιού στο H, η ιδιότητα ότι κάθε ζεύγος κορυφών του G ανήκει σε κάποιον κύκλο διατηρείται, άρα το γράφημα G που προκύπτει ότι είναι επίσης 2-συνεκτικό. □

Πρόταση 11 (Whitney). *Ένα γράφημα G είναι 2-συνεκτικό αν και μόνο αν είναι το G είναι κύκλος ή το G προκύπτει από έναν κύκλο με διαδοχικές προσθήκες μονοπατιών.*

Με άλλα λόγια, αν θέλουμε να κατασκευάζουμε 2-συνεκτικά γραφήματα, ξεκινάμε από έναν κύκλο και κάνουμε διαδοχικά προσθήκες μονοπατιών.

Παράδειγμα, ο κύβος Q_3 (ο οποίος είναι 3-συνεκτικός, άρα και 2-συνεκτικός) μπορεί να κατασκευασθεί από τον κύκλο C_4 με την επόμενη ακολουθία από προσθήκες μονοπατιών.



Διαισθητικά οι κορυφές ενός k-συνεκτικού γραφήματος πρέπει να έχουν αρκετά μεγάλο βαθμό (τουλάχιστον k) και το γράφημα να περιέχει αρκετούς δεσμούς ώστε να υπάρχουν τουλάχιστον k ανεξάρτητα μονοπάτια μεταξύ οποιουδήποτε ζεύγους κορυφών του. Η επόμενη πρόταση δίνει ένα ελάχιστο φράγμα για τον αριθμό των δεσμών που πρέπει να έχει ένα γράφημα για να είναι k-συνεκτικό.

Πρόταση 12. *Έστω G ένα k-συνεκτικό γράφημα με n κορυφές. Τότε ο αριθμός των δεσμών του G είναι τουλάχιστον* $\left\lceil \frac{kn}{2} \right\rceil$.

Απόδειξη. Σε κάθε k-συνεκτικό γράφημα ισχύει ότι $\delta(G) \geq k$. Πράγματι, αν υπάρχει κορυφή v με $d(v) < k$, τότε μπορούμε να διαχωρίσουμε την κορυφή v από τις κορυφές του γραφήματος σβήνοντας λιγότερες από k κορυφές, τους γείτονες της, το οποίο είναι άτοπο.

Κατά συνέπεια ισχύει ότι

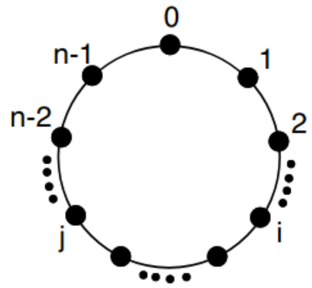
$$2|E| = \sum_{v\in V} d(v) \geq \sum_{v\in V} \delta(G) = n\delta(G) \geq nk. \qquad \square$$

Για παράδειγμα, ένα 5-συνεκτικό γράφημα με 31 κορυφές πρέπει να έχει τουλάχιστον $\left\lceil \frac{5\cdot 31}{2} \right\rceil = \lceil 77.5 \rceil = 78$ δεσμούς.

Άσκηση 3. *Να δειχθεί ότι δεν υπάρχει 3-συνεκτικό γράφημα με 75 κορυφές και 112 δεσμούς.*

Ο Frank Harary έδωσε έναν αλγόριθμο για την κατασκευή ενός k-συνεκτικού γραφήματος $H_{k,n}$ με n κορυφές το οποίο έχει ακριβώς $\left\lceil \frac{kn}{2} \right\rceil$ δεσμούς.

Η κατασκευή ξεκινά με τον κύκλο C_n, του οποίου οι κορυφές αριθμούνται διαδοχικά (με την φορά του ρολογιού).

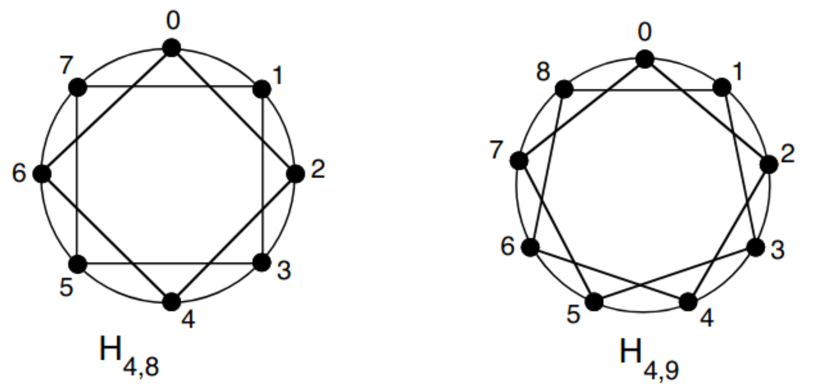


Η κατασκευή του $H_{n,k}$ εξαρτάται από την αρτιότητα των k και n. Διακρίνουμε τρεις περιπτώσεις:

- k άρτιος.
  Έστω $k = 2r$, τότε οι κορυφές i, j ενώνονται με δεσμό ανν τα σημεία απέχουν στον κύκλο απόσταση το πολύ r δηλαδή

$$|j - i| \leq r \text{ ή/και } n - |j - i| \leq r$$

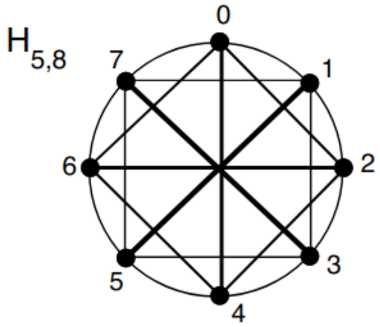
  Για παράδειγμα,



  Το γράφημα $H_{2r,n}$ έχει ακριβώς rn δεσμούς. Πράγματι, κάθε κορυφή συνδέεται με $2r$ κορυφές, οπότε $2|E| = 2rn$. Άρα, το $H_{2r,n}$ έχει τον ζητούμενο αριθμό δεσμών αφού $rn = \frac{kn}{2} = \left\lceil \frac{kn}{2} \right\rceil$ (kn άρτιος).

- k περιττός και n άρτιος.
Έστω $k = 2r + 1$. Αρχικά κατασκευάζουμε το γράφημα $H_{2r,n}$ και προσθέτουμε τις $\frac{n}{2}$ "διαμέτρους" του αρχικού κύκλου C_n, δηλαδή τους δεσμούς που ενώνουν τις κορυφές i και $i + \frac{n}{2}$, για $i = 0, \dots, \frac{n}{2} - 1$.

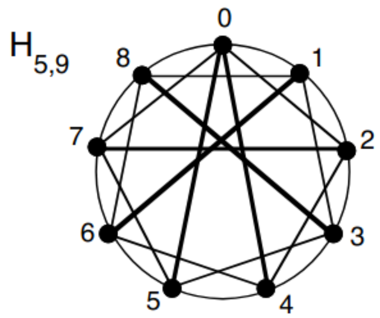
  Για παράδειγμα,



  Ο συνολικός αριθμός των δεσμών του $H_{2r+1,n}$ είναι ίσος με $rn + \frac{n}{2} = \frac{(2r+1)n}{2} = \frac{kn}{2} = \left\lceil \frac{kn}{2} \right\rceil$ (kn άρτιος).

- Τα k, n είναι περιττά.
Έστω $k = 2r+1$. Αρχικά κατασκευάζουμε το γράφημα $H_{2r,n}$ και προσθέτουμε τις $\frac{n+1}{2}$ "οιονεί-διαμέτρους" του αρχικού κύκλου C_n, δηλαδή τους δεσμούς που ενώνουν τις κορυφές 0 και $\frac{n-1}{2}$, τις κορυφές 0 και $\frac{n+1}{2}$ και τους δεσμούς που ενώνουν τις κορυφές i και $i + \frac{n+1}{2}$, για $i = 1, \dots, \frac{n-3}{2}$.

  Για παράδειγμα,



  Ο συνολικός αριθμός των δεσμών του $H_{2r+1,n}$ είναι ίσος με $rn + \frac{n+1}{2} = \frac{(2r+1)n+1}{2} = \frac{kn+1}{2} = \left\lceil \frac{kn}{2} \right\rceil$ (kn περιττός).

Πρόταση 13. *Το γράφημα του Harary $H_{k,n}$ είναι k-συνεκτικό (με τον ελάχιστο αριθμό δεσμών μεταξύ όλων των γραφημάτων με n κορυφές).*

Άσκηση 4. *Να κατασκευασθούν τα γραφήματα $H_{3,7}$, $H_{4,7}$, $H_{5,7}$.*

3.2. Διαμερίσεις γραφημάτων. Ένα σημαντικό πρόβλημα στις εφαρμογές των γραφημάτων είναι η εύρεση μιας διαμέρισης των κορυφών ενός γραφήματος έτσι ώστε οι κορυφές που περιέχονται σε διαφορετικά σύνολα της διαμέρισης να έχουν όσο το δυνατόν λιγότερες συνδέσεις (δεσμούς), σε αντίθεση με τις κορυφές που ανήκουν στην ίδια κλάση.

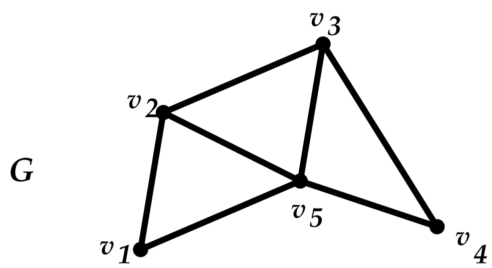
Για παράδειγμα, στην περίπτωση όπου ένα γράφημα αποτελείται από 3 συνεκτικές συνιστώσες μια τέτοια ιδανική διαμέριση με 3 κλάσεις είναι η διαμέριση των κορυφών σύμφωνα με τις συνεκτική συνιστώσα που ανήκουν.

Στην περίπτωση όπου το γράφημα είναι συνεκτικό ή όπου θέλουμε να βρούμε περισσότερες κλάσεις από ότι οι συνεκτικές συνιστώσεις χρειαζόμαστε ένα μέτρο που να αξιολογεί το πόσο καλή είναι μια διαμέριση σε σύγκριση με μια άλλη διαμέριση.

Έστω $G = (V, E)$ ένα γράφημα δεσμών και S, T δύο ξένα υποσύνολα κορυφών του V. Ο **όγκος** (volume) του S ορίζεται ως ο αριθμός των δεσμών του G που έχουν τουλάχιστον ένα άκρο στο S και συμβολίζεται με $\mathrm{vol}(S)$.

Επίσης, η **τομή** (cut) των S, T ορίζεται ως ο αριθμός των δεσμών των οποίων το ένα άκρο ανήκει στο S και το άλλο άκρο στο T και συμβολίζεται με $\mathrm{cut}(S, T)$.

Για παράδειγμα, έστω $S = \{v_1, v_2\}$ και $T = \{v_3, v_5, v_4\}$ μια διαμέριση των κορυφών του παρακάτω γραφήματος σε δύο σύνολα.



Τότε $\mathrm{vol}(S) = 4$, $\mathrm{vol}(T) = 6$ και $\mathrm{cut}(S, T) = 3$.

Οι "καλές" διαμερίσεις έχουν μεγάλο όγκο για κάθε κλάση της διαμέρισης και μικρές τομές μεταξύ όλων των ζευγών κλάσεων.

Συγκεκριμένα, ορίζουμε την **κανονικοποιημένη τομή** (normalized cut) των S, T ως εξής:

$$\mathrm{ncut}(S, T) = \frac{\mathrm{cut}(S, T)}{\mathrm{vol}(S)} + \frac{\mathrm{cut}(S, T)}{\mathrm{vol}(T)}$$

Στο πρόβλημα της διαμέρισης ο στόχος είναι εύρεση συνόλων S, T που ελαχιστοποιούν την παράσταση $\mathrm{ncut}(S, T)$.

Για την διαμέριση του παραπάνω παραδείγματος, έχουμε ότι $\mathrm{ncut}(S, T) = \frac{3}{4} + \frac{3}{6} = 1.25$.

Ενώ, για την διαμέριση $S' = \{v_2, v_3\}$ και $T' = \{v_1, v_5, v_4\}$ έχουμε $\mathrm{vol}(S') = 5$, $\mathrm{vol}(T') = 6$, $\mathrm{cut}(S', T') = 4$, οπότε $\mathrm{ncut}(S', T') = \frac{4}{5} + \frac{4}{6} = 1.46667$. Επομένως, η προηγούμενη διαμέριση είναι καλύτερη από αυτήν.

Η μετρική ncut δεν ευνοεί την επιλογή συνόλων με λίγες κορυφές. Για παράδειγμα, η διαμέριση $S'' = \{v_2\}$ και $T'' = \{v_1, v_3, v_4, v_5\}$ δίνει $\mathrm{vol}(S'') = 3$, $\mathrm{vol}(T'') = 7$, $\mathrm{cut}(S'', T'') = 3$, οπότε $\mathrm{ncut}(S'', T'') = \frac{3}{3} + \frac{3}{7} = 1.42857$.

Η εύρεση της βέλτιστης διαμέρισης με k κλάσεις γίνεται αναδρομικά. Αρχικά, βρίσκεται η βέλτιστη διαμέριση με δύο κλάσεις. Στην συνέχεια, διαμερίζονται (εκλεπτύνονται) αναδρομικά οι δύο κλάσεις σε μικρότερες με κριτήριο την ελαχιστοποίηση της κανονικοποιημένης τομής, μέχρι να κατασκευασθούν οι ζητούμενες k κλάσεις.

Οι γέφυρες ενός γραφήματος (εφόσον υπάρχουν) μπορούν να αποτελέσουν ένα ευρετικό κριτήριο για να την δημιουργία της αρχικής διαμέρισης. Υπάρχουν αλγόριθμοι που υπολογίζουν βέλτιστες διαμερίσεις με την βοήθεια τεχνικών γραμμικής άλγεβρας και πιο συγκεκριμένα υπολογίζοντας τις ιδιοτιμές και τα ιδιοδιανύσματα μιας συγκεκριμένης μήτρας που κωδικοποιεί το γράφημα και ονομάζεται Λαπλασιανή (Laplacian matrix).

Πρόταση 14. $\sum_{i=1}^{|V|} d(v_i) = 2|E|$.

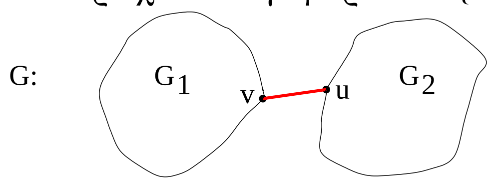
Απόδειξη: Κάθε δεσμός του γραφήματος συνεισφέρει κατά 2 στο άθροισμα των βαθμών (λόγω των άκρων του). Άρα, οι $|E|$ δεσμοί που περιέχει το γράφημα θα δημιουργούν συνολικό άθροισμα βαθμών $2|E|$. □

Πόρισμα 15. *Σε κάθε γράφημα ο αριθμός κόμβων με περιττό βαθμό είναι άρτιος.*

Απόδειξη: Έστω $v_1, v_2, \ldots, v_{2k+1}$ οι (περιττοί σε πλήθος) κόμβοι με περιττό βαθμό. Τότε το άθροισμα των βαθμών των κόμβων αυτών θα ήταν περιττό (έστω Π) ως άθροισμα περιττού πλήθους περιττών αριθμών. Δεδομένου ότι το άθροισμα των βαθμών των κόμβων με άρτιο βαθμό είναι άρτιο (έστω A) ως άθροισμα άρτιων αριθμών, το συνολικό άθροισμα των βαθμών του γραφήματος θα ήταν $\Pi + A$: περιττός, το οποιο σύμφωνα με την Πρόταση 1 είναι άτοπο. Άρα το πλήθος των κόμβων με περιττό βαθμό είναι άρτιο. □

Πρόταση 16. *Αν ένα συνεκτικό γράφημα δεσμών G περιέχει μόνο κορυφές με άρτιο βαθμό, τότε δεν περιέχει γέφυρες.*

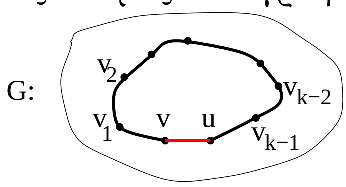
Απόδειξη. Έστω ότι το G περιέχει την γέφυρα $e = \{v, u\}$.



Αν διαγράψουμε από το G την γέφυρα e, στο νέο γράφημα $G - e$ οι κορυφές v, u θα έχουν περιττούς βαθμούς και θα προκύψουν δύο συνεκτικές συνιστώσες G_1, G_2 στις οποίες θα ανήκουν αντίστοιχα τα v, u. Τότε όμως όλες οι υπόλοιπες κορυφές της G_1 θα έχουν άρτιο βαθμό, επομένως το άθροισμα των βαθμών των κορυφών της G_1 θα είναι περιττό, άτοπο. Άρα, το G δεν περιέχει γέφυρα. □

Πρόταση 17. *Αν ένα συνεκτικό γράφημα δεσμών G δεν περιέχει γέφυρα, τότε κάθε κορυφή του ανήκει πάνω σε κάποιο κύκλο.*

Απόδειξη. Έστω $e = \{v, u\}$ ένας δεσμός του γραφήματος.



Αφού ο δεσμός e δεν είναι γέφυρα το γράφημα $G - e$ είναι συνεκτικό, άρα υπάρχει μονοπάτι $P = (v, v_1, v_2, \ldots, v_k = u$ στο $G - e$ που συνδέει τις κορυφές v και u. Επομένως, στο G η v ανήκει στον κύκλο $(v, v_1, v_2, \ldots, u, v)$. □

5. ΓΡΑΦΗΜΑ EULER - ΓΡΑΦΗΜΑ HAMILTON

ΓΡΑΦΗΜΑ EULER

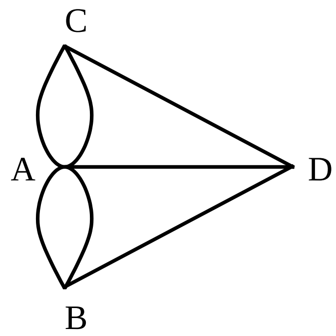
Αν υπάρχει (τουλάχιστον) ένας δρόμος του γραφήματος G, ο οποίος χρησιμοποιεί όλους τους δεσμούς του G, λέγεται **δρόμος Euler**. Αν το G περιέχει ένα κλειστό δρόμο Euler, τότε λέγεται **γράφημα Euler**.

Πρόταση 18.

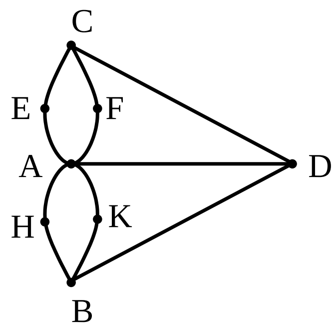
i) Ένα συνεκτικό γράφημα G περιέχει τουλάχιστον ένα δρόμο Euler αν και μόνο αν περιέχει το πολύ δύο κόμβους περιττού βαθμού. Αν περιέχει δύο τέτοιους κόμβους v_1, v_2, τότε όλοι οι δρόμοι Euler του G είναι $v_1 - v_2$ δρόμοι.

ii) Ένα συνεκτικό γράφημα G είναι γράφημα Euler αν και μόνο αν όλοι οι κόμβοι του έχουν άρτιο βαθμό, στην περίπτωση αυτή, όλοι οι δρόμοι Euler του G είναι κλειστοί.

Παρατήρηση: Η Πρόταση 18 i) δίνει και την (αρνητική) απάντηση στο πρόβλημα των γεφυρών του Königsberg, που παρουσιάστηκε στην εισαγωγή, αφού το γράφημα

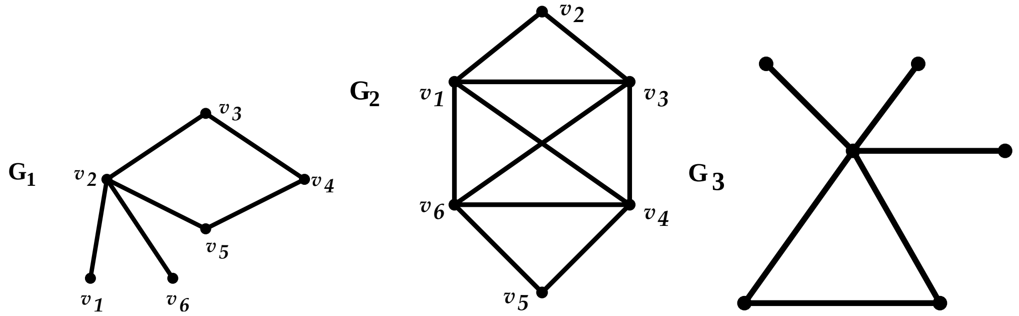


ή, αυστηρότερα, το γράφημα



έχει πάνω από δύο κόμβους περιττού βαθμού.

Παραδείγματα:



Το γράφημα G_1 περιέχει το δρόμο Euler

$$(v_1, v_2, v_3, v_4, v_5, v_2, v_6)$$

αλλά δεν είναι γράφημα Euler, (αφού περιέχει και κόμβους περιττού βαθμού).

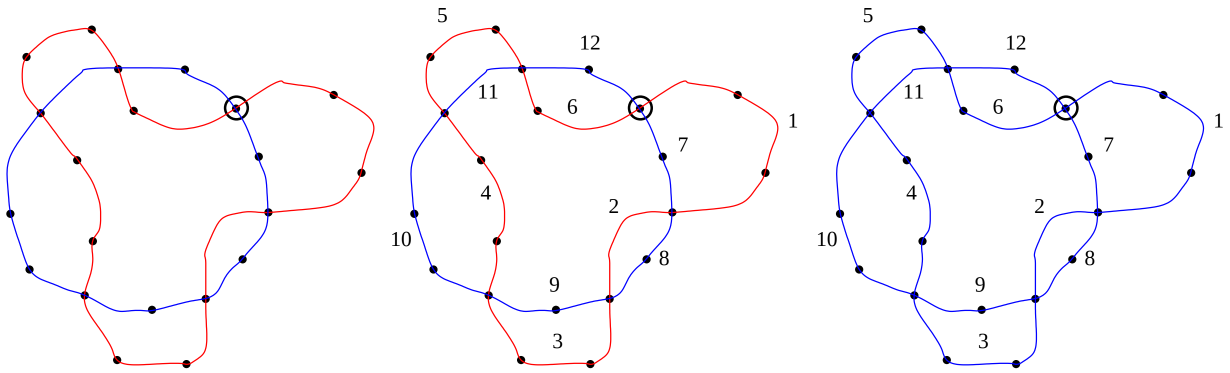
Το γράφημα G_2 (του οποίου όλοι οι κόμβοι έχουν άρτιο βαθμό) είναι γράφημα Euler, αφού περιέχει τον κλειστό δρόμο Euler

$$(v_6, v_1, v_2, v_3, v_4, v_1, v_3, v_6, v_4, v_5, v_6).$$

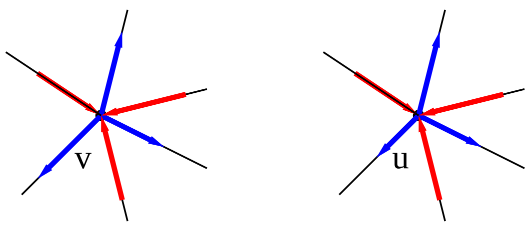
Το γράφημα G_3 δεν περιέχει δρόμο Euler, (αφου περιέχει πάνω από δύο κόμβους περιττού βαθμού.

Για την εύρεση ενός κλειστού δρόμου Euler μπορεί να χρησιμοποιηθεί ο επόμενος αλγόριθμος. Ο αλγόριθμος στηρίζεται σε δύο βασικές ιδέες:

- Αν έχουμε δύο κλειστούς δρόμους που διέρχονται από μια ή περισσότερες κοινές κορυφές και χρησιμοποιούν διαφορετικούς δεσμούς, τότε μπορούμε να τους ενώσουμε σε ένα μεγαλύτερο κλειστό δρόμο.



- Επειδή όλες οι κορυφές έχουν άρτιο βαθμό, κάθε δρόμος που ξεκινά από μια οποιαδήποτε κορυφή v μπορεί πάντα να επιστρέψει στην αρχική κορυφή v ανεξάρτητα από τις επιλογές που γίνονται κατά την διάτρεξή του. Ο λόγος είναι ότι για κάθε δεσμό που μας απομακρύνει από την αρχική v υπάρχει τουλάχιστον ένας δεσμός που μας οδηγεί πάλι σ' αυτή. Αντίθετα σε οποιαδήποτε άλλη κορυφή u δεν μπορούμε να φτάσουμε σε αδιέξοδο αφού για κάθε δεσμό που μας οδηγεί στην u υπάρχει τουλάχιστον ένας δεσμός που μας απομακρύνει από την u.



Αλγόριθμος του Hierholzer

Είσοδος: Ένα γράφημα Euler G.

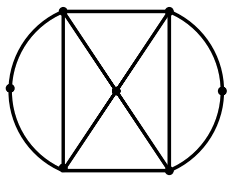
Έξοδος: Ένας κλειστός δρόμος Euler.

Βήμα 1 Επιλέγουμε οποιαδήποτε κορυφή v η οποία είναι άκρο δεσμού που δεν έχουμε διασχίσει. Κατασκευάζουμε έναν κλειστό δρόμο W που περιέχει την v επιλέγοντας αυθαίρετα έναν οποιονδήποτε από τους δεσμούς που δεν έχουμε ήδη διασχίσει μέχρι να επιστρέψουμε και πάλι στην v.

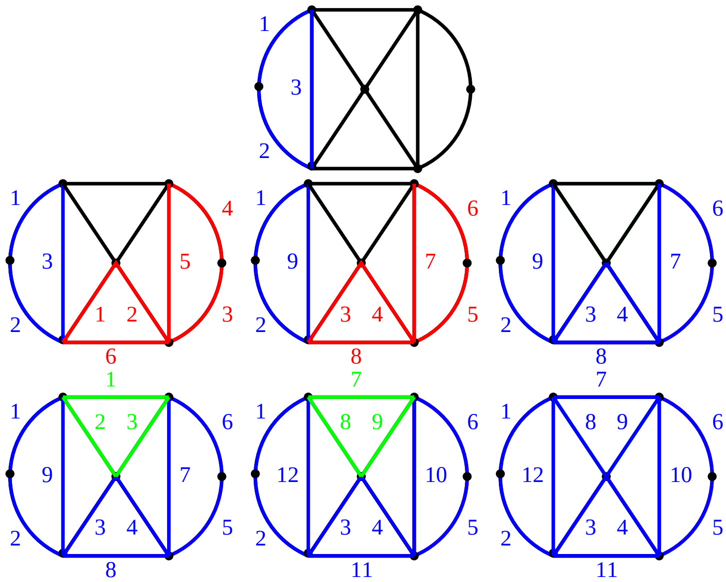
Βήμα 2 Αν υπάρχει κορυφή u η οποία είναι άκρο δεσμού που δεν έχουμε συμπεριλάβει στον κλειστό δρόμο W, επαναλαμβάνουμε το Βήμα 1 για την κορυφή u και **ενώνουμε** τους δύο κλειστούς δρόμους που προκύπτουν.

Αφού το G είναι συνεκτικό επαναλαμβάνοντας τα βήματα 1 και 2 θα εξαντλήσουμε όλους τους δεσμούς του γραφήματος και θα δημιουργήσουμε ένα κλειστό δρόμο Euler για το G.

Παράδειγμα: Να βρεθεί ένας κλειστός δρόμος Euler για το γράφημα



Λύση.

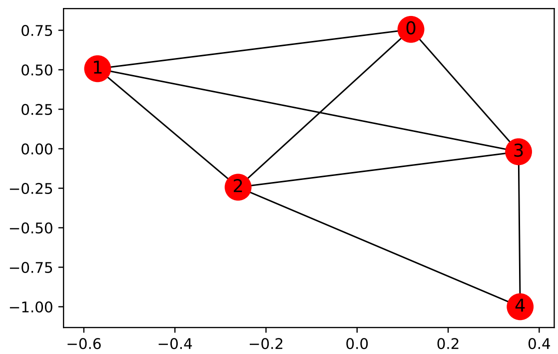


□

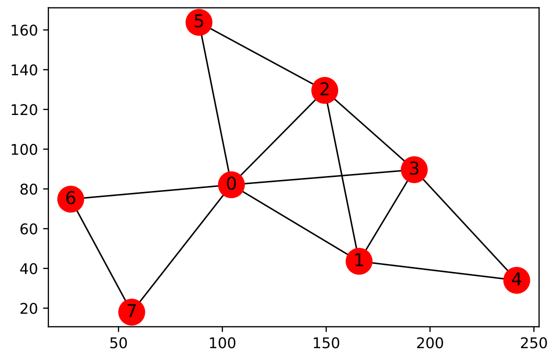
Μπορούμε να βρούμε ένα κλειστό δρόμο Euler σε ένα (συνεκτικό) γράφημα με άρτιους βαθμούς κορυφών χρησιμοποιώντας την μέθοδο `euleriean_circuit(G)`:

```
import networkx as nx
import matplotlib.pyplot as plt

H = nx.house_x_graph()
nx.draw_networkx(H)
plt.show()
if nx.is_eulerian(H):
    W1 = nx.eulerian_circuit(H)
    print("An Eulerian circuit for H:", list(W1))
else:
    print("H is not Eulerian graph")

Seq = [6,4,4,4,2,2,2,2]
G = nx.havel_hakimi_graph(Seq)
pos = nx.drawing.nx_agraph.graphviz_layout(G)
nx.draw_networkx(G,pos)
plt.savefig("euler_example0.eps")
if nx.is_eulerian(G):
    #W is an Eulerian circuit for G
    W = nx.eulerian_circuit(G)
    print("An Eulerian circuit for G:", list(W))
    #print("An Eulerian circuit for G:", [u for u, v in W])
    i = 1
    for e in W:
        G1 = G.subgraph(e)
        nx.draw_networkx_edges(G1,pos,edge_color='blue',width=3.0)
        nx.draw_networkx_nodes(G1,pos,node_color='blue',width=3.0)
        plt.savefig("euler_example"+str(i)+".eps")
        nx.draw_networkx_edges(G1,pos,edge_color='red',width=3.0)
        nx.draw_networkx_nodes(G1,pos,node_color='red',width=3.0)
        i += 1
    plt.savefig("euler_example"+str(i)+".eps")
else:
    print("G is not Eulerian graphs")
plt.show()
```

Output:

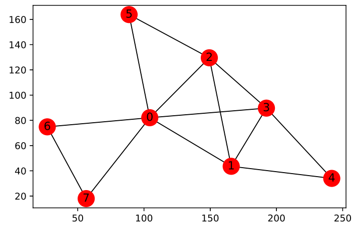



```
H is not Eulerian graph
```

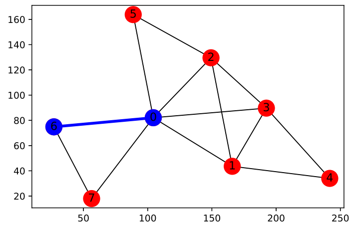



```
An Eulerian circuit for G: [(0, 6), (6,
  7), (7, 0), (0, 5), (5, 2), (2, 3),
  (3, 4), (4, 1), (1, 2), (2, 0), (0, 1)
  , (1, 3), (3, 0)]
```

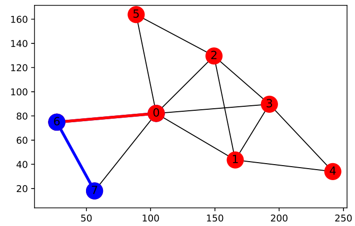
Στο επόμενο παράδειγμα απεικονίζεται η εκτέλεση του αλγορίθμου του Heirholzer πάνω στο γράφημα του προηγούμενου σχήματος. Προσοχή! Η σειρά εκτέλεσης έχει απεικονισθεί βουστροφηδόν.



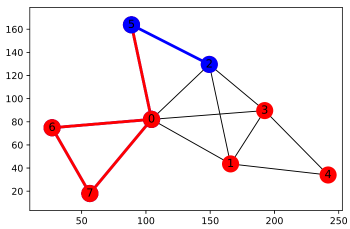
1 →



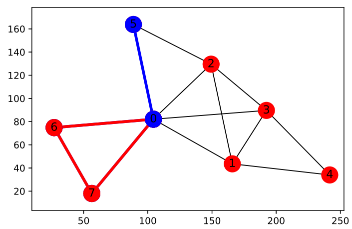
2 →



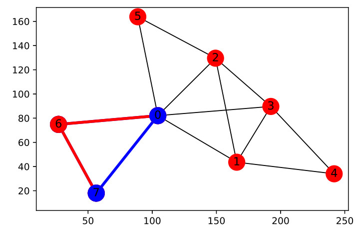
3 ↓



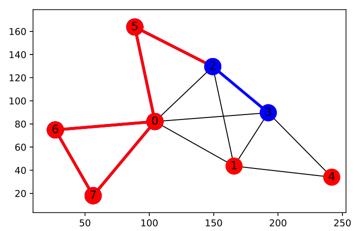
↓ 6 ←



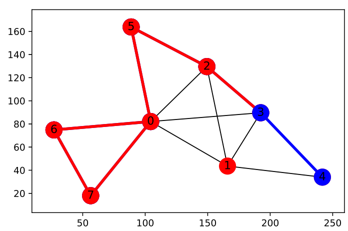
5 ←



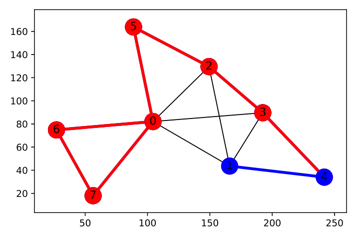
4 (Βρέθηκε ο 1ος κύκλος)



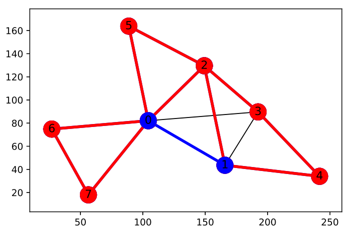
7 →



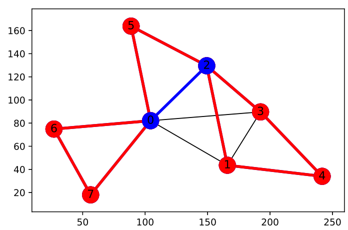
8 →



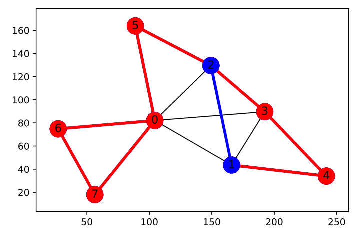
9 ↓



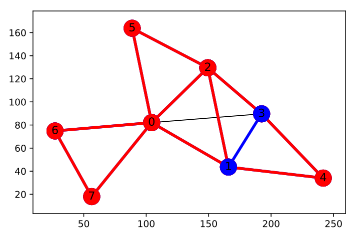
↓ 12 ←



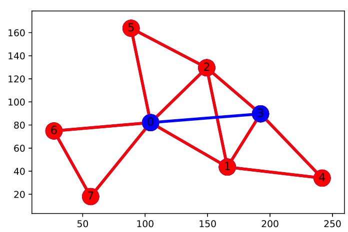
11 (Βρέθηκε 2ος κύκλος) ←



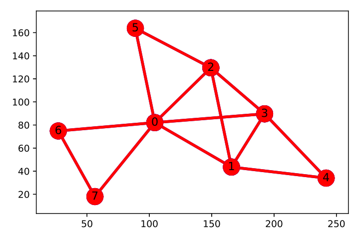
10



13 →



14 (Βρέθηκε 3ος κύκλος) →



15

Ένας εναλλακτικός αλγόριθμος για την εύρεση του δρόμου Euler είναι ο επόμενος:

Αλγόριθμος του Fleury

Είσοδος: Συνεκτικό γράφημα δεσμών G.

Έξοδος: Ένας (κλειστός) δρόμος Euler W (αν υπάρχει).

Παρατηρήσεις: Κατά την εκτέλεση του αλγορίθμου μετακινούμαστε από κορυφή σε κορυφή. Η κορυφή στην οποία βρισκόμαστε ονομάζεται τρέχουσα κορυφή. Κάθε δεσμός που διασχίζουμε διαγράφεται από το γράφημα G.

Βήμα 0 Αν το G έχει πάνω από 2 κορυφές περιττού βαθμού, τότε το G δεν περιέχει δρόμο Euler. Επιστρέφουμε τον κενό δρόμο W.

Αν το G περιέχει ακριβώς 2 κορυφές v, u περιττού βαθμού, τότε ξεκινάμε από την μία από αυτές. Αλλιώς, ξεκινάμε από οποιαδήποτε κορυφή v του G.

Θέτουμε $W = v$.

Βήμα 1 Από την τρέχουσα κορυφή v επιλέγουμε να διασχίσουμε έναν από τους γειτονικούς δεσμούς της $\{v, u\}$ αρκεί να μην είναι γέφυρα στο γράφημα G, εκτός αν δεν υπάρχει άλλος διαθέσιμος δεσμός.

Βήμα 2 Διαγράφουμε από το G τον δεσμό $\{v, u\}$ που διασχίζουμε, θέτουμε ως τρέχουσα κορυφή την νέα κορυφή u, προσθέτουμε στο τέλος του W την u.

Βήμα 3 Αν το γράφημα G δεν είναι μηδενικό (δηλαδή περιέχει δεσμούς που δεν έχουμε διασχίσει) εκτελούμε το Βήμα 1.

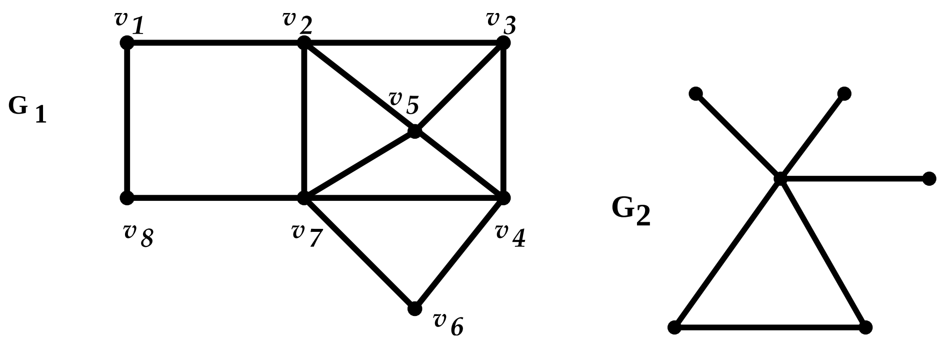
Αλλιώς, ο αλγόριθμος τερματίζει και ο W είναι ένας (κλειστός) δρόμος Euler για το γράφημα G.

Το υπολογιστικά βαρύτερο κομμάτι του αλγορίθμου είναι ο έλεγχος για το πότε ένας δεσμός του γραφήματος είναι γέφυρα.

ΓΡΑΦΗΜΑ HAMILTON

Ένας κύκλος του G ο οποίος διέρχεται από όλους τους κόμβους του G λέγεται **κύκλος Hamilton**. Αν το G περιέχει ένα κύκλο Hamilton, λέγεται **γράφημα Hamilton**.

Παραδείγματα:



Το γράφημα G_1 είναι γράφημα Hamilton, αφού περιέχει τον κύκλο Hamilton $(v_1, v_2, v_5, v_3, v_4, v_6, v_7, v_8, v_1)$, ενώ το γράφημα G_2 δεν είναι γράφημα Hamilton, αφού προφανώς δεν περιέχει ένα κύκλο Hamilton.

Παρατήρηση: Μέχρι σήμερα δεν έχει βρεθεί μια αναγκαία και ικανή συνθήκη για το πότε ένα γράφημα περιέχει κύκλο Hamilton. Επισης, η εύρεση ενός κύκλου Hamilton σε ένα τυχαίο γράφημα θεωρείται δύσκολο πρόβλημα. (Συγκεκριμένα, ανήκει στην κατηγορία των NP-complete προβλημάτων.)

Γνωστά αποτελέσματα:

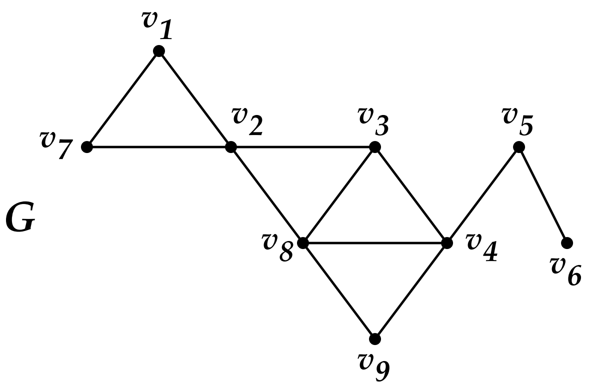
Πρόταση 19 (Dirac 1952). *Αν σε ένα απλό γράφημα G, με $|V(G)| = n \geq 3$, ισχύει ότι $d(v) + d(u) \geq n$ για κάθε ζεύγος v, u μη γειτονικών κορυφών, τότε το G είναι γράφημα Hamilton.*

Πρόταση 20 (Pósa 1962). *Αν σε ένα απλό γράφημα G, με $|V(G)| = n \geq 3$ και ακολουθία βαθμών $d_1 \leq d_2 \leq \cdots \leq d_n$, ισχύει ότι $d_i \geq i + 1$ για κάθε $i < n/2$, τότε το G είναι γράφημα Hamilton.*

Πρόταση 21 (Chvátal 1972). *Αν σε ένα απλό γράφημα G, με $|V(G)| = n \geq 3$ και ακολουθία βαθμών $d_1 \leq d_2 \leq \cdots \leq d_n$, ισχύει ότι $d_i \geq i + 1$ ή/και $d_{n-i} \geq n - i$ για κάθε $i < n/2$, τότε το G είναι γράφημα Hamilton.*

Ασκήσεις προς επίλυση

(1) Δίδεται το γράφημα G



Να ορισθούν:

i) Μια διαδρομή μήκους 8 από το v_1 στο v_3.

ii) Ένας δρόμος μήκους 5 από το v_3 στο v_8.

iii) Ένα μονοπάτι μήκους 4 από το v_2 στο v_3.

iv) Μια κλειστή διαδρομή μήκους 6 (που να μην είναι δρόμος).

v) Ένας κλειστός δρόμος μήκους 6 (που να μην είναι κύκλος).

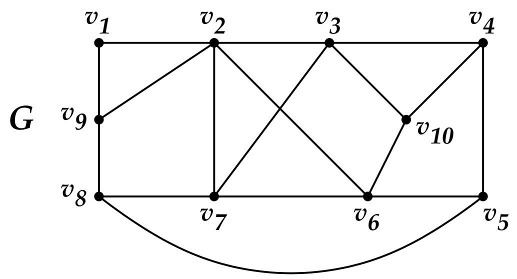
vi) Ένας κύκλος μήκους 5.

vii) Ένα άκυκλο υπογράφημά του H με $V(H) = V(G)$.

viii) Να ευρεθούν (αν υπάρχουν) οι κλειδώσεις και οι ισθμοί του.

ix) Να ευρεθούν τα μπλοκ του.

(2) Δίδεται το γράφημα G



i) Να εξετασθεί αν είναι μη διαχωρίσιμο.

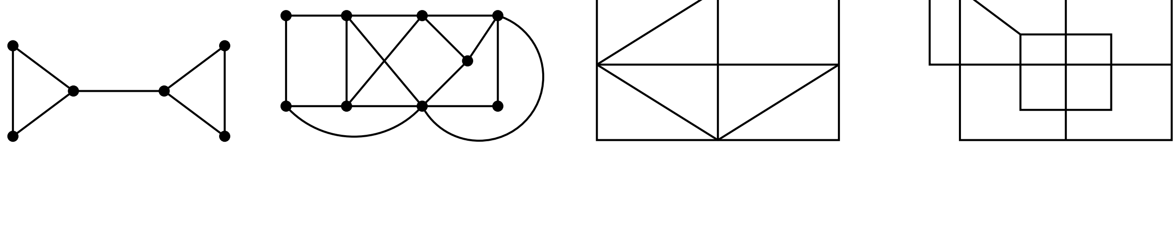
ii) Να ευρεθεί ένα σύνολο κλειδώσεών του.

(3) Για καθένα από τα παρακάτω γραφήματα, να εξετασθεί:

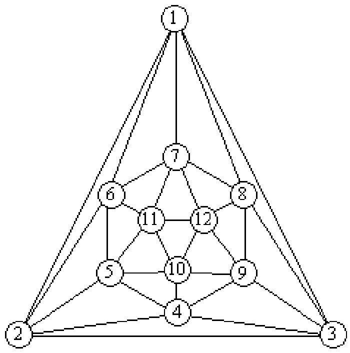
i) Αν υπάρχει δρόμος Euler.

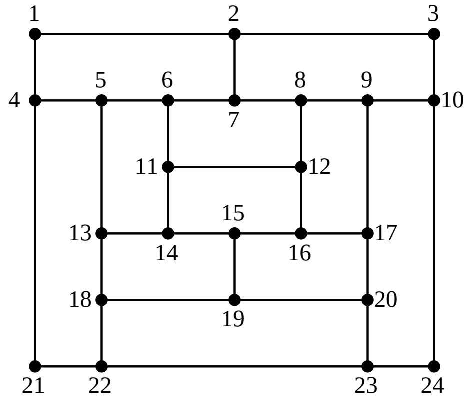
ii) Αν είναι γράφημα Euler.

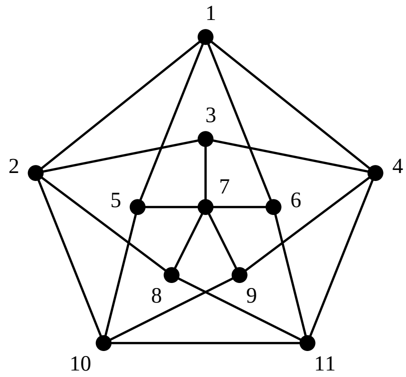
Στην περίπτωση όπου υπάρχει (κλειστός) δρόμος Euler να βρεθεί ένας τέτοιος (κλειστός) δρόμος.



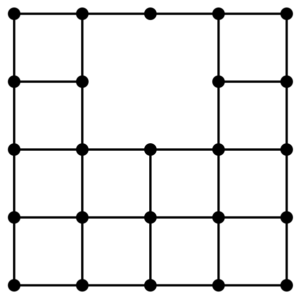
(4) Να βρεθεί ένας κύκλος Hamilton για τα παρακάτω γραφήματα:







(5) **(*)** Να δειχθεί ότι το παρακάτω γράφημα δεν είναι γράφημα Hamilton.



(6) **(*)** Έστω G ένα κανονικό γράφημα με άρτιο αριθμό κορυφών. Να δειχθεί ότι ένα τουλάχιστον από τα G, G^c είναι γράφημα Hamilton.

(7) Να εξετασθεί αν υπάρχουν γραφήματα με τις παρακάτω ακολουθίες βαθμών

α) $(3, 3, 3, 3, 3, 3, 3, 3, 3)$

β) $(5, 5, 5, 5, 5, 5, 5, 5, 5, 2, 2, 2)$

(8) **(*)** Να κατασκευασθεί ένα 3-συνεκτικό γράφημα με 8 κορυφές και 12 δεσμούς.

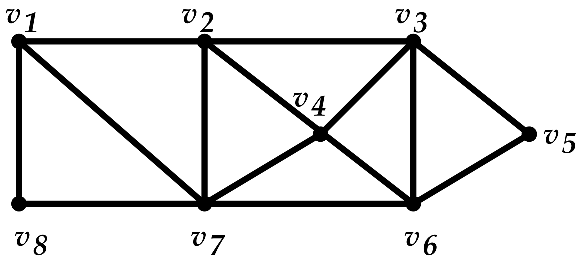
6. ΑΠΟΣΤΑΣΕΙΣ

Απόσταση (distance) $d(u,v)$ μεταξύ δύο κόμβων u, v μιας συνιστώσας του G ονομάζεται το ελάχιστο μήκος μεταξύ όλων των διαδρομών που τους συνδέουν.

Μερικοί συγγραφείς γενικεύουν τον παραπάνω ορισμό, ορίζοντας ως απόσταση δύο κόμβων που ανήκουν σε διαφορετικές συνιστώσες ενός μη συνεκτικού γραφήματος το ∞.

Γεωδαιτικό ή **συντομότερο** λέγεται κάθε $u - v$ μονοπάτι ενός γραφήματος G, με μήκος ίσο με $d(u,v)$.

Παράδειγμα:



$d(v_1, v_4) = 2$.

(v_1, v_2, v_4): γεωδαιτικό, (v_1, v_7, v_4): γεωδαιτικό.

(v_1, v_8, v_7, v_4): όχι γεωδαιτικό.

Η εύρεση της απόστασης ανάμεσα σε δύο κόμβους u, v ενός γραφήματος μπορεί να γίνει χρησιμοποιώντας την αναζήτηση σε πλάτος, ή τους αλγορίθμους του Dijkstra, ή των Bellman - Ford. (Οι δύο τελευταίοι αλγόριθμοι μπορούν να δώσουν απαντήσεις και σε γραφήματα που έχουν βάρη πάνω στους δεσμούς τους.)

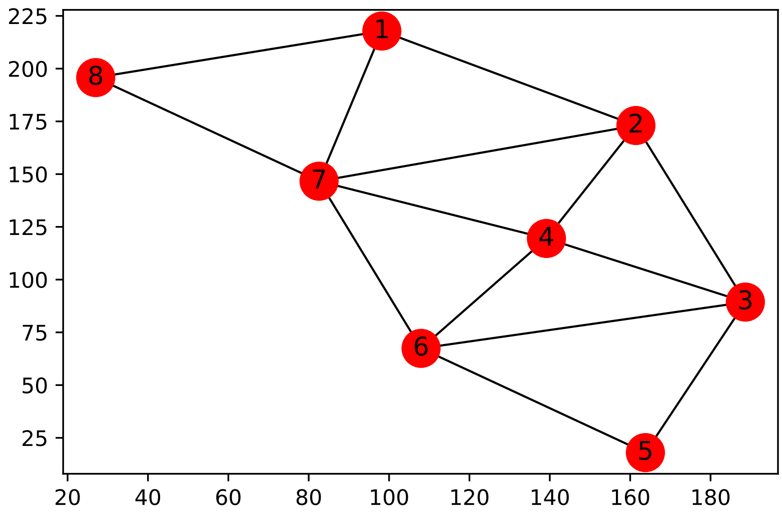
Η βιβλιοθήκη `networkx` έχει τις μεθόδους `shortest_path(G,v,u)` και `shortest_path_length(G,v,u)` που υπολογίζουν ένα γεωδαιτικό μονοπάτι μεταξύ των κορυφών v και u και την απόσταση των v και u αντίστοιχα.

Επιπρόσθετα, υπάρχουν οι μέθοδοι `all_pairs_shortest_path(G)` και `all_pairs_shortest_path_length(G)` που υπολογίζουν ένα γεωδαιτικό μονοπάτι ανάμεσα σε όλα τα ζεύγη κορυφών και τα αντίστοιχα μήκη τους.

```
import networkx as nx
import matplotlib.pyplot as plt
import numpy as np

G = nx.Graph()
V = [1,2,3,4,5,6,7,8]
E = [[1,2],[1,7],[1,8],[2,3],[2,4],[2,7],[3,4],[3,5],[3,6],
     [4,6],[4,7],[5,6],[6,7],[7,8]]

G.add_nodes_from(V)
G.add_edges_from(E)
pos = nx.nx_agraph.graphviz_layout(G)
nx.draw_networkx(G,pos)

v, u = 1, 4
print("The distance between nodes",v,"and",u,"is:",nx.shortest_path_length(G,v,u))

shortestpaths = dict(nx.all_pairs_shortest_path(G))
for v in G:
```

```
    print("A shortest path between",v)
    for u in G:
        print("and",u,"is:",shortestpaths[v][u])

alldistances = dict(nx.all_pairs_shortest_path_length(G))
n = G.order()
D = np.zeros((n,n)) #create a n x n matrix with zero entries
for v in G:
    for u in G:
        #D has numbering 0...7 while nodes are numbered 1...8
        D[u-1][v-1] = alldistances[v][u]
print("The distances between all nodes pairs of G are:")
print(D)

plt.show()
```

Output:




```
The distance between nodes
   1 and 4 is: 2
A shortest path between 1
and 1 is: [1]
and 2 is: [1, 2]
and 3 is: [1, 2, 3]
and 4 is: [1, 2, 4]
and 5 is: [1, 2, 3, 5]
and 6 is: [1, 7, 6]
and 7 is: [1, 7]
and 8 is: [1, 8]
A shortest path between 2
and 1 is: [2, 1]
and 2 is: [2]
and 3 is: [2, 3]
and 4 is: [2, 4]
and 5 is: [2, 3, 5]
and 6 is: [2, 3, 6]
and 7 is: [2, 7]
and 8 is: [2, 1, 8]
A shortest path between 3
and 1 is: [3, 2, 1]
and 2 is: [3, 2]
and 3 is: [3]
and 4 is: [3, 4]
and 5 is: [3, 5]
and 6 is: [3, 6]
and 7 is: [3, 2, 7]
and 8 is: [3, 2, 1, 8]
A shortest path between 4
and 1 is: [4, 2, 1]
and 2 is: [4, 2]
and 3 is: [4, 3]
and 4 is: [4]
and 5 is: [4, 3, 5]
and 6 is: [4, 6]
and 7 is: [4, 7]
and 8 is: [4, 7, 8]
A shortest path between 5
and 1 is: [5, 3, 2, 1]
and 2 is: [5, 3, 2]
and 3 is: [5, 3]
and 4 is: [5, 3, 4]
and 5 is: [5]
and 6 is: [5, 6]
and 7 is: [5, 6, 7]
and 8 is: [5, 6, 7, 8]
A shortest path between 6
and 1 is: [6, 7, 1]
and 2 is: [6, 3, 2]
and 3 is: [6, 3]
and 4 is: [6, 4]
and 5 is: [6, 5]
and 6 is: [6]
and 7 is: [6, 7]
and 8 is: [6, 7, 8]
A shortest path between 7
and 1 is: [7, 1]
and 2 is: [7, 2]
and 3 is: [7, 2, 3]
and 4 is: [7, 4]
and 5 is: [7, 6, 5]
and 6 is: [7, 6]
```

```
and 7 is: [7]
and 8 is: [7, 8]
A shortest path between 8
and 1 is: [8, 1]
and 2 is: [8, 1, 2]
and 3 is: [8, 1, 2, 3]
and 4 is: [8, 7, 4]
and 5 is: [8, 7, 6, 5]
and 6 is: [8, 7, 6]
and 7 is: [8, 7]
and 8 is: [8]
```

```
The distances between all nodes pairs of G are:
[[0. 1. 2. 2. 3. 2. 1. 1.]
 [1. 0. 1. 1. 2. 2. 1. 2.]
 [2. 1. 0. 1. 1. 1. 2. 3.]
 [2. 1. 1. 0. 2. 1. 1. 2.]
 [3. 2. 1. 2. 0. 1. 2. 3.]
 [2. 2. 1. 1. 1. 0. 1. 2.]
 [1. 1. 2. 1. 2. 1. 0. 1.]
 [1. 2. 3. 2. 3. 2. 1. 0.]]
```

Μερικές φορές θέλουμε να αποκτήσουμε μια συνολική εικόνα για τις πιθανές αποστάσεις που έχουν τα ζεύγη κόμβων του γραφήματος.

Εκκεντρότητα (eccentricity) $e(v)$ ενός κόμβου v ενός συνεκτικού γραφήματος G είναι η $\max_{u \in V(G)} d(u, v)$, (δηλαδή η μεγαλύτερη απόσταση που έχει ο v από όλες τις άλλες κορυφές u).

Διάμετρος (diameter) $d(G)$ ενός συνεκτικού γραφήματος G λέγεται το μήκος του μεγαλύτερου γεωδαιτικού του, (δηλαδή η μεγαλύτερη απόσταση ανάμεσα σε όλα τα δυνατά ζεύγη κόμβων).

Παρατήρηση: Προφανώς $d(G) = \max_{v \in V(G)} e(v)$.

Ακτίνα (radius) $r(G)$ ενός συνεκτικού γραφήματος G είναι η ελάχιστη εκκεντρότητα, ανάμεσα σε όλους τους κόμβους του G, δηλαδή $r(G) = \min_{v \in V(G)} e(v)$.

Ο v λέγεται **κεντρικός κόμβος** (central node) του συνεκτικού γραφήματος G, αν $e(v) = r(G)$. **Κέντρο** (center) του συνεκτικού γραφήματος ονομάζεται το σύνολο των κεντρικών του κόμβων.

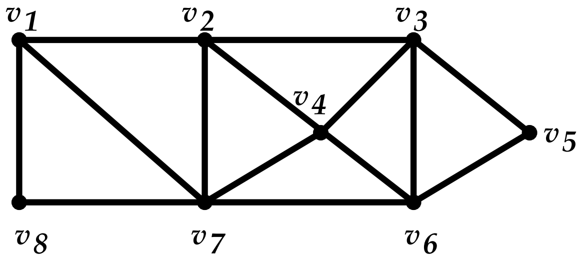
Ο v λέγεται **περιφερειακός κόμβος** (peripheral node) του συνεκτικού γραφήματος G, αν $e(v) = d(G)$. **Περιφερειακό σύνολο** (periphery set) του συνεκτικού γραφήματος ονομάζεται το σύνολο των περιφερειακών του κόμβων.

Για κάθε κόμβο v συμβολίζουμε με $s(v)$ το συνολικό άθροισμα όλων των αποστάσεών του από κάθε άλλο κόμβο u, δηλαδή

$$s(v) = \sum_{u \in G} d(u, v)$$

Βαρύκεντρο (barycenter ή median) ενός γραφήματος ονομάζεται το σύνολο των κόμβων v που ελαχιστοποιούν το συνολικό άθροισμα $s(v)$ όλων των αποστάσεων τους από κάθε άλλο κόμβο u.

Παράδειγμα:



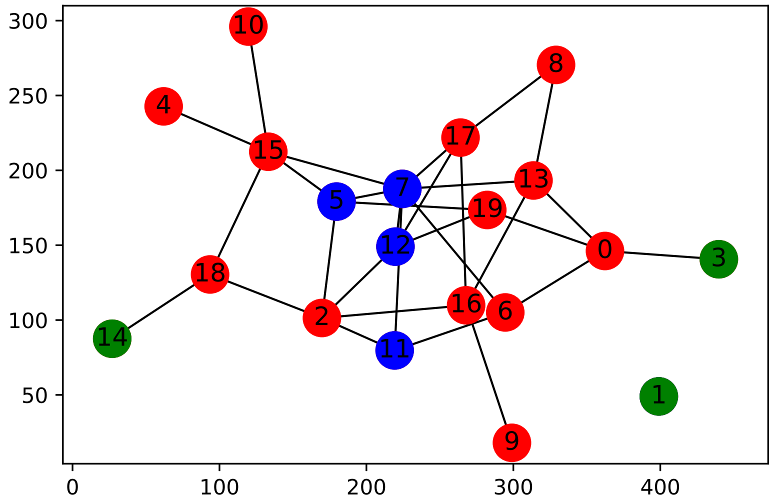
$d(G) = 3.$
$e(v_1) = e(v_3) = e(v_5) = e(v_8) = 3.$
$e(v_2) = e(v_4) = e(v_6) = e(v_7) = 2.$
$r(G) = 2.$
Κέντρο του $G = \{v_2, v_4, v_6, v_7\}$.
Περιφερειακό σύνολο του $G = \{v_1, v_3, v_5, v_8\}$.
$s(v_5) = s(v_8) = 14,\ s(v_1) = 12,\ s(v_3) = 11,\ s(v_2) = s(v_4) = s(v_6) = 10,\ s(v_7) = 9.$
Βαρύκεντρο του $G = \{v_7\}$.

Η βιβλιοθήκη `networkx` διαθέτει τις μεθόδους `radius(G)`, `diameter(G)`, `center(G)` και `periphery(G)` για τον υπολογισμό των αντίστοιχων εννοιών. Υπολογιστικά όμως συμφέρει να υπολογίσουμε μια φορά τις εκκεντρότητες των κορυφών του G με την μέθοδο `eccentricity(G)` και έπειτα, με βάση αυτές, τα υπόλοιπα στατιστικά. Επίσης, διαθέτει την μέθοδο `barycenter(G)` για τον υπολογισμό του βαρύκεντρου του G.

```
import networkx as nx
import matplotlib.pyplot as plt

#create a random graph with n nodes and m edges
n,m = 20,30
G = nx.gnm_random_graph(n,m)
pos = nx.nx_agraph.graphviz_layout(G)
nx.draw_networkx(G,pos)

print("G has",nx.number_connected_components(G),"connected component(s)")

#for every connected component compute the eccentricities of its nodes
#and then its radius, diameter and center
Gcc = nx.connected_components(G)
for cc in Gcc:
    G1 = G.subgraph(cc)        #G1 is a connected component
    #find the eccentricities every node in G1
    eccdict = nx.eccentricity(G1)
    #print the eccentricities of every node in G1
    for v in G1:
        print("The eccentricity of node",v,"is",eccdict[v])

    all_values = eccdict.values()
    G1rad = min(all_values) #compute the radius of G1
    G1diam = max(all_values) #compute the diameter of G1
    #find center of G1
    G1center = []
```

```
    G1periphery = []
    for node in eccdict:
        eccnode = eccdict[node]
        if eccnode == G1rad:
            G1center.append(node)
        if eccnode == G1diam:
            G1periphery.append(node)
    print("The connected component",cc,"has radius",G1rad,"diameter",G1diam,"center",
   G1center,"and periphery",G1periphery)

    #color blue the central and peripheral nodes of G1
    G2 = G.subgraph(G1center)  #G2 is the induced subgraph of Gcenter
    nx.draw_networkx_nodes(G2,pos,node_color='blue',width=3.0)
    G3 = G.subgraph(G1periphery)  #G2 is the induced subgraph of Gcenter
    nx.draw_networkx_nodes(G3,pos,node_color='green',width=3.0)

plt.show()
```

Output:

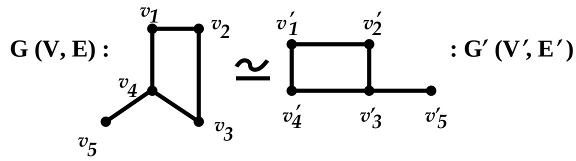



```
G has 2 connected component(s)
The eccentricity of node 0 is 5
The eccentricity of node 2 is 4
The eccentricity of node 3 is 6
The eccentricity of node 4 is 5
The eccentricity of node 5 is 3
The eccentricity of node 6 is 4
The eccentricity of node 7 is 3
The eccentricity of node 8 is 5
The eccentricity of node 9 is 5
The eccentricity of node 10 is 5
The eccentricity of node 11 is 3
The eccentricity of node 12 is 3
The eccentricity of node 13 is 4
The eccentricity of node 14 is 6
The eccentricity of node 15 is 4
The eccentricity of node 16 is 4
The eccentricity of node 17 is 4
The eccentricity of node 18 is 5
The eccentricity of node 19 is 4
The connected component {0, 2, 3, 4, 5,
    6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
    16, 17, 18, 19} has radius 3 diameter
    6 center [5, 7, 11, 12] and periphery
    [3, 14]
The eccentricity of node 1 is 0
The connected component {1} has radius 0
    diameter 0 center [1] and periphery
    [1]
```

7. ΙΣΟΜΟΡΦΑ ΓΡΑΦΗΜΑΤΑ

Τα γραφήματα δεσμών $G = (V, E)$ και $G' = (V', E')$ ονομάζονται **ισόμορφα** αν και μόνο αν υπάρχει αμφιμονοσήμαντη απεικόνιση $f : V \to V'$, με $\{x, y\} \in E \Leftrightarrow \{f(x), f(y)\} \in E'$. Αν δύο γραφήματα G και G' είναι ισόμορφα, θα γράφουμε $G \simeq G'$.

Παραδείγματα: Τα επόμενα γραφήματα είναι ισόμορφα:



διότι για την $f : V \to V'$ με

$$\begin{aligned} f(v_1) &= v'_4, \\ f(v_2) &= v'_1, \\ f(v_3) &= v'_2, \\ f(v_4) &= v'_3, \\ f(v_5) &= v'_5, \end{aligned}$$

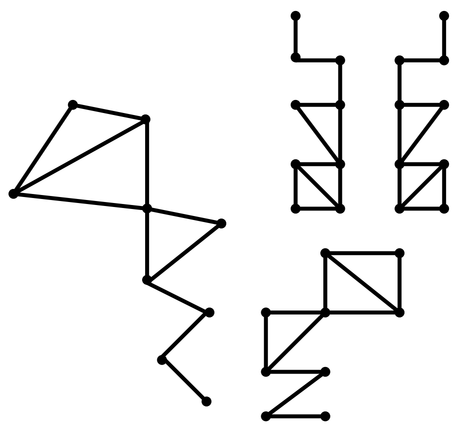
έχουμε πράγματι ότι

$$\{v_i, v_j\} \in E \Leftrightarrow \{f(v_i), f(v_j)\} \in E',$$

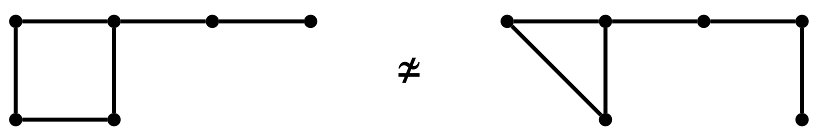
(για παράδειγμα:

$$\begin{aligned} &\{v_1, v_2\} \in E \text{ και } \{v'_4, v'_1\} \in E', \\ &\{v_2, v_4\} \notin E \text{ και } \{v'_1, v'_3\} \notin E', \\ &\{v_4, v_5\} \in E \text{ και } \{v'_3, v'_5\} \in E', \quad \text{κ.ο.κ.}). \end{aligned}$$

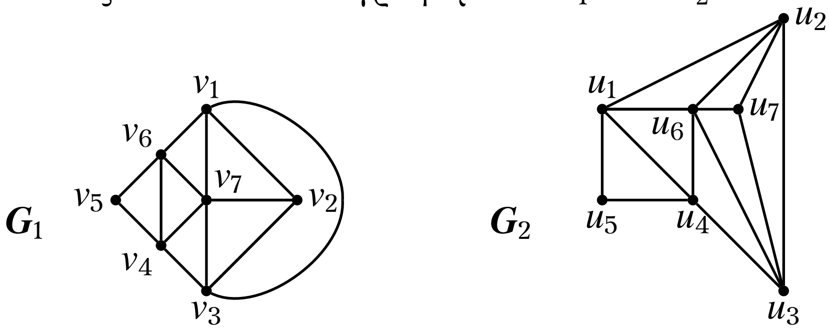
Τα επόμενα γραφήματα είναι όλα ισόμορφα:



Αντίθετα τα επόμενα δύο γραφήματα δεν είναι ισόμορφα:



Παράδειγμα Να εξετασθεί αν τα γραφήματα G_1 και G_2 είναι ισόμορφα.



Λύση. Τα G_1, G_2 είναι ισόμορφα. Ένας ισομορφισμός f είναι ο εξής:

$$f(v_1) = u_2$$
$$f(v_2) = u_7$$
$$f(v_3) = u_3$$
$$f(v_4) = u_4$$
$$f(v_5) = u_5$$
$$f(v_6) = u_1$$
$$f(v_7) = u_6$$

□

Μπορούμε να ελέγξουμε αν τα G_1, G_2 είναι ισόμορφα χρησιμοποιώντας την μέθοδο `GraphMatcher` της βιβλιοθήκης networkx.

```
import networkx as nx
G1 = nx.Graph()
G1.add_nodes_from(range(1,8))
G1.add_edges_from([[1,2],[1,3],[1,6],[1,7],[2,3],[2,7],
                   [3,4],[3,7],[4,5],[4,6],[4,7],[5,6],[6,7]])
G2 = nx.Graph()
G2.add_nodes_from(range(1,8))
G2.add_edges_from([[1,2],[1,5],[1,4],[1,6],[2,3],[2,6],
                   [2,7],[3,4],[3,6],[3,7],[4,5],[4,6],[6,7]])
#Test whether the graphs are isomorphic
GM = nx.isomorphism.GraphMatcher(G1,G2)
if GM.is_isomorphic():  #If G1, G2 are isomorphic
    print("The graphs are isomorphic")
    print("An isomorphism between them is the following:")
    for i in G1:
        print("v",i,"->","u",GM.mapping[i])
    print(GM.mapping)
else:
    print("The graphs are not isomorphic")
```

Output:

```
The graphs are isomorphic
An isomorphism between them is the following:
v 1 -> u 3
v 2 -> u 7
v 3 -> u 2
v 4 -> u 1
v 5 -> u 5
v 6 -> u 4
```

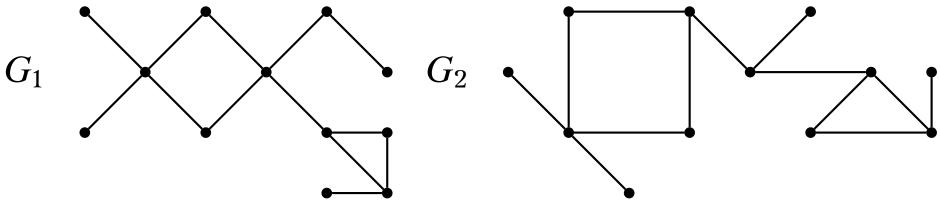
```
v 7 -> u 6
{4: 1, 3: 2, 1: 3, 6: 4, 5: 5, 7: 6, 2: 7}
```

Παρατήρηση Παρατηρήστε ότι ο αλγόριθμος ανακάλυψε έναν διαφορετικό ισομορφισμό από αυτόν που δόθηκε αρχικά.

Πρόταση 22. *Αν δύο γραφήματα G, H είναι ισόμορφα, τότε:*
i) Έχουν την ίδια ακολουθία βαθμών, και μάλιστα ισχύει ότι $d_G(v) = d_H(f(v))$, για κάθε $v \in V(G)$.
ii) Έχουν ισόμορφα υπογραφήματα.

Συνήθως η Πρόταση 22 χρησιμοποιείται αρνητικά, δηλαδή αν δύο γραφήματα δεν έχουν την ίδια ακολουθία βαθμών, ή/και δεν έχουν τα ίδια υπογραφήματα, τότε δεν είναι ισόμορφα. Για παράδειγμα,

(1)



$(4, 4, 3, 3, 2, 2, 2, 2, 1, 1, 1, 1)$ $\qquad$ $(4, 3, 3, 3, 3, 2, 2, 2, 1, 1, 1, 1)$

$G_1 \not\cong G_2$, διότι έχουν διαφορετικές ακολουθίες βαθμών.

(2)

Λύση. Δεν είναι ισόμορφα: Το G_1 περιέχει κύκλο μήκους 3 ενώ το G_2 όχι. □

Πάλι, μπορούμε να ελέγξουμε αν τα G_1, G_2 είναι ισόμορφα χρησιμοποιώντας την μέθοδο `GraphMatcher` της βιβλιοθήκης networkx.

```python
import networkx as nx
G1 = nx.Graph()
G1.add_nodes_from(range(1,8))
G1.add_edges_from([[1,2],[1,7],[2,3],[2,6],[3,4],[3,5],[4,5],[5,6],[6,7]])
G2 = nx.Graph()
G2.add_nodes_from(range(1,8))
G2.add_edges_from([[1,2],[1,4],[1,7],[2,3],[3,4],[3,5],[4,6],[5,6],[6,7]])
#Test whether the graphs are isomorphic
GM = nx.isomorphism.GraphMatcher(G1,G2)
if GM.is_isomorphic():  #If G1, G2 isomorphic? then
    print("The graphs are isomorphic")
    print("An isomorphism between them is the following:")
    for i in G1:
        print("v",i,"->","u",GM.mapping[i])
    print(GM.mapping)
```

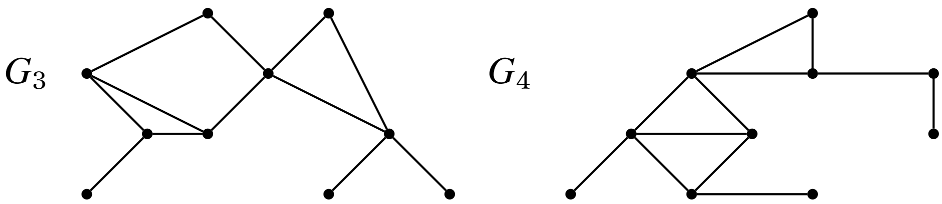
```
else:
     print("The graphs are not isomorphic")
     print("Degree sequence of G1:",sorted((d for n, d in G1.degree()),reverse=
    True))
     print("Degree sequence of G2:",sorted((d for n, d in G2.degree()),reverse=
    True))
```

Output:

```
The graphs are not isomorphic
Degree sequence of G1: [3, 3, 3, 3, 2, 2, 2]
Degree sequence of G2: [3, 3, 3, 3, 2, 2, 2]
```

Παρατήρηση Παρατηρήστε ότι παρόλο που τα γραφήματα είναι μη ισόμορφα έχουν την ίδια ακολουθία βαθμών. Στην περίπτωση όπου τα δύο γραφήματα δεν είναι ισόμορφα, η μέθοδος `GraphMatcher` δεν μας δίνει κάποια εξήγηση γιατί συμβαίνει αυτό.

(3)



$(4, 4, 3, 3, 3, 2, 2, 1, 1, 1)$ $(4, 4, 3, 3, 3, 2, 2, 1, 1, 1)$

$G_3 \not\cong G_4$, διότι, ενώ έχουν την ίδια ακολουθία βαθμών, έχουν διαφορετικά υπογραφήματα, (για παράδειγμα, το G_3 περιέχει δύο K_3, ενώ το G_4 περιέχει τρία K_3).

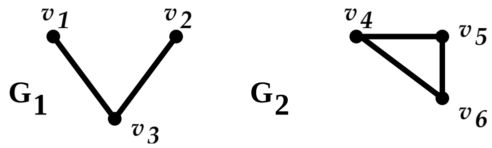
Παρατήρηση. Ο έλεγχος αν δύο μεγάλα γραφήματα είναι ισόμορφα ή όχι είναι υπολογιστικά δύσκολος μέχρι σήμερα. Παρόλα αυτά υπάρχει η άποψη το πρόβλημα μπορεί να λυθεί γρηγορότερα αλλά δεν έχει βρεθεί ακόμα η κατάλληλη ιδέα.

8. ΠΡΑΞΕΙΣ

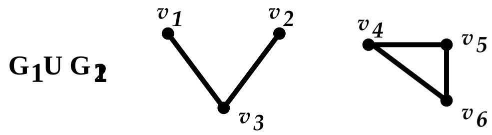
Έστω $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$.

Ένωση $G = G_1 \cup G_2$ είναι το γράφημα $G = (V, E)$ με $V = V_1 \cup V_2$ και $E = E_1 \cup E_2$.

Παράδειγμα: Έστω

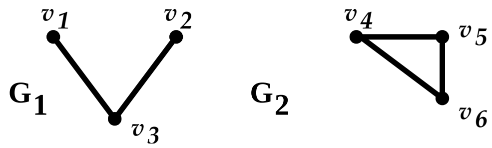


Η ένωση των γραφημάτων G_1 και G_2 είναι το γράφημα $G = G_1 \cup G_2$:

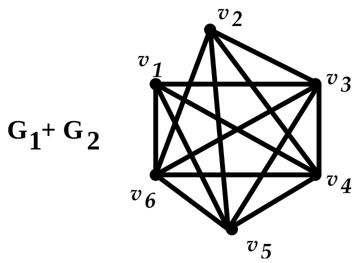


Άθροισμα $G = G_1 + G_2$ είναι το γράφημα $G = (V, E)$ με $V = V_1 \cup V_2$ και $E = E_1 \cup E_2 \cup \{\{v_i, v_j\} : v_i \in V_1, v_j \in V_2\}$ (δηλαδή το $G_1 + G_2$ είναι το $G_1 \cup G_2$ μαζί με όλους τους δεσμούς που ενώνουν τα στοιχεία του V_1 με στοιχεία του V_2).

Παράδειγμα: Έστω



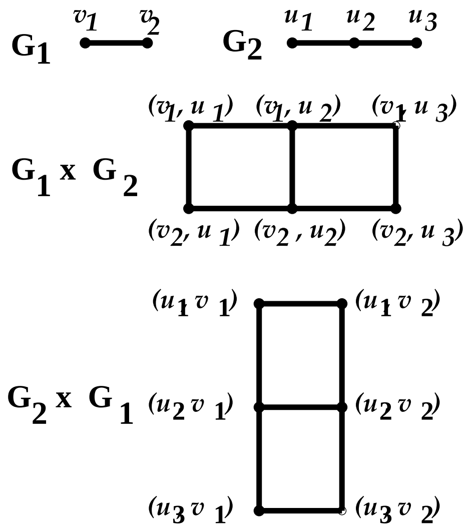
Το άθροισμα των γραφημάτων G_1 και G_2 είναι το γράφημα $G = G_1 + G_2$:



Γινόμενο $G = G_1 \times G_2$ είναι το γράφημα $G = (V, E)$ με $V = V_1 \times V_2$ και αν $\alpha = (v_1, u_1)$, $\beta = (v_2, u_2) \in V$ τότε $\{\alpha, \beta\} \in E$ αν και μόνο αν:

$$(v_1 = v_2 \text{ και } \{u_1, u_2\} \in E(G_2)), \text{ ή } (u_1 = u_2 \text{ και } \{v_1, v_2\} \in E(G_1)).$$

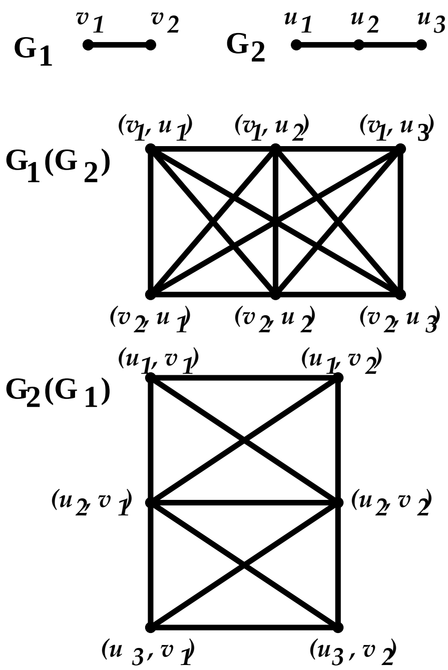
Παράδειγμα:



Σύνθεση $G = G_1(G_2)$ είναι το γράφημα $G = (V, E)$ με $V = V_1 \times V_2$ και αν $\alpha = (v_1, u_1)$, $\beta = (v_2, u_2) \in V$ τότε $\{\alpha, \beta\} \in E$ αν και μόνο αν:

$$(\{v_1, v_2\} \in E(G_1)), \text{ ή } (v_1 = v_2 \text{ και } \{u_1, u_2\} \in E(G_2)).$$

Παράδειγμα:

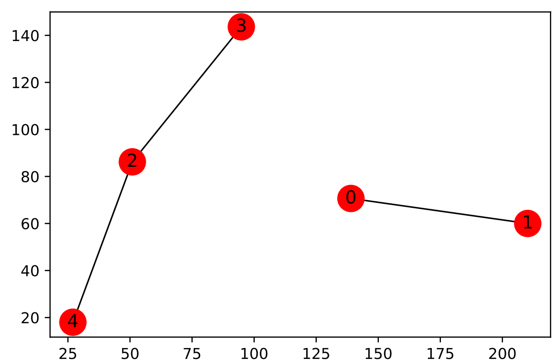


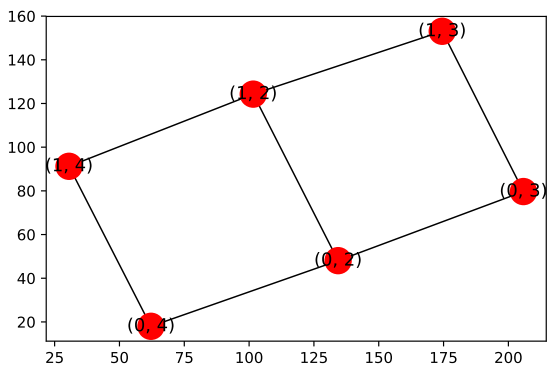
Μπορούμε να χρησιμοποιήσουμε την βιβλιοθήκη networkx για να υλοποιήσουμε τις παραπάνω πράξεις.

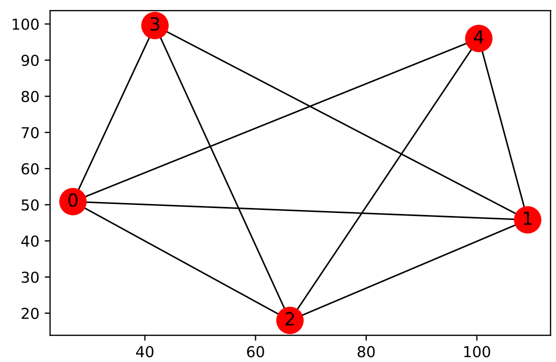
```
import networkx as nx
import matplotlib.pyplot as plt

G1 = nx.Graph()
V1 = [0,1]
E1 = [[0,1]]
G1.add_nodes_from(V1)
G1.add_edges_from(E1)

```

```
G2 = nx.Graph()
V2 = [2,3,4]
E2 = [[2,3],[2,4]]
G2.add_nodes_from(V2)
G2.add_edges_from(E2)

#The disjoint union works even if V1 and V2 are not disjoint
#It renumbers the vertices starting from G1 and then continuing
#with G2
Gunion = nx.disjoint_union(G1, G2)
pos = nx.nx_agraph.graphviz_layout(Gunion)
nx.draw_networkx(Gunion,pos)
plt.show()

Gproduct = nx.cartesian_product(G1,G2)
pos = nx.nx_agraph.graphviz_layout(Gproduct)
nx.draw_networkx(Gproduct,pos)
plt.show()

def graphsum(G1, G2):
    Gsum = nx.disjoint_union(G1,G2)
    for i in range(G1.order()):
        for j in range(G1.order(), G1.order() + G2.order()):
            Gsum.add_edge(i, j)
    return Gsum
Gsum = graphsum(G1, G2)
pos = nx.nx_agraph.graphviz_layout(Gsum)
nx.draw_networkx(Gsum,pos)
plt.show()
```

Output:







9. ΑΝΑΠΟΦΕΥΚΤΕΣ ΙΔΙΟΤΗΤΕΣ (*)

Complete disorder is impossible. – Theodore S. Motzkin (1908 – 1970).

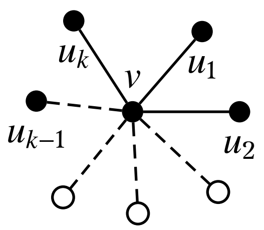
Στα γραφήματα δεσμών και ιδιαίτερα στα μεγάλα γραφήματα εμφανίζονται πάντα ή σχεδόν πάντα ορισμένα χαρακτηριστικά που εκ πρώτης όψεως μοιάζουν με συμπτώσεις. Υπάρχει μια ολόκληρη κατηγορία αποτελεσμάτων που δείχνουν ότι όσο τυχαίο να είναι ένα γράφημα πάντα ή σχεδόν πάντα θα εμφανίζει κάποια "κανονικότητα", πάντα ή σχεδόν πάντα θα περιέχει κάποιες "συμπτώσεις".

Στην ενότητα αυτή δίνονται δύο πολύ απλά παραδείγματα τέτοιων αποτελεσμάτων που ισχύουν πάντα.

Πρόταση 23. *Σε κάθε συνεκτικό γράφημα δεσμών $G = (V, E)$ με $|V| \geq 2$, υπάρχουν τουλάχιστον δύο κορυφές με τον ίδιο βαθμό οι οποίες είναι γειτονικές ή έχουν κοινό γείτονα.*

Λύση. Έστω $\Delta(G) = k$ ο μέγιστος βαθμός των κορυφών του γραφήματος και έστω $v \in V$ μια κορυφή με βαθμό k.

Η v έχει k γειτονικές κορυφές $u_1, u_2, \ldots, u_k$ και κάθε μία από αυτές έχει βαθμό από 1 μέχρι k.

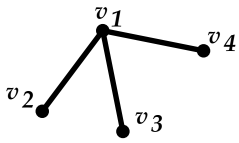


Διακρίνουμε δύο περιπτώσεις:

- Αν κάποια από τις u_1, u_2, …, u_k έχει βαθμό k, τότε το γράφημα περιέχει δύο γειτονικές κορυφές με τον ίδιο βαθμό.
- Αν καμία από τις u_1, u_2, …, u_k δεν έχει βαθμό k, τότε υπάρχουν $k-1$ δυνατές τιμές βαθμών για τους k γείτονες της v, οπότε από την αρχή του περιστερεώνα, δύο τουλάχιστον από αυτές έχουν τον ίδιο βαθμό και κοινό γείτονα την v. □

Πρόταση 24. *Αν $|V| \geq 6$, τότε ή το G ή το G^c περιέχει τουλάχιστον ένα υπογράφημα ισόμορφο με το K_3, (δηλαδή ένα τρίγωνο).*

Απόδειξη. Έστω $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ και έστω ότι ο v_1 είναι ενωμένος με τουλάχιστον τρεις κόμβους στο G. Έστω λοιπόν: $\{v_1, v_2\}$, $\{v_1, v_3\}$, $\{v_1, v_4\} \in E$.



Αν $\{v_2, v_3\} \in E$ τότε οι v_1, v_2, v_3 είναι κορυφές τρίγωνου στο G.

Όμοια αν $\{v_2, v_4\} \in E$, ή $\{v_3, v_4\} \in E$.

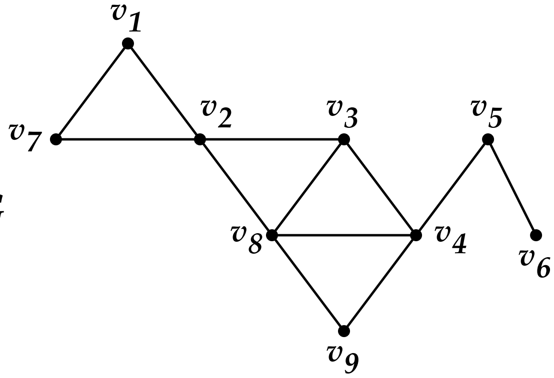
Αν τώρα $\{v_2, v_3\}$, $\{v_2, v_4\}$, $\{v_3, v_4\} \notin E$, τότε θα έχουμε $\{v_2, v_3\}$, $\{v_2, v_4\}$, $\{v_3, v_4\} \in E^c$ και άρα οι v_2, v_3, v_4 είναι κορυφές τρίγωνου στο G^c.

Αν τέλος ο v_1 είναι ενωμένος με λιγότερους από τρεις κόμβους στο G, θα είναι ενωμένος με τουλάχιστον τρεις κόμβους στο G^c. Τότε παίρνουμε το αντίστοιχο αποτέλεσμα, ξεκινώντας από το G^c. □

Ασκήσεις προς επίλυση

(1) Δίδεται το γράφημα G

G



Να ευρεθούν:

i) Η απόσταση $d(v_2, v_5)$.

ii) Δύο γεωδεσικά ανάμεσα στους κόμβους v_2 και v_4.

iii) Η διάμετρος $d(G)$.

iv) Η ακτίνα $r(G)$.

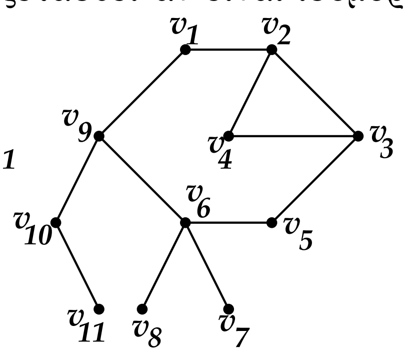
v) Το κέντρο του G.

vi) Το περιφερειακό σύνολο του G.

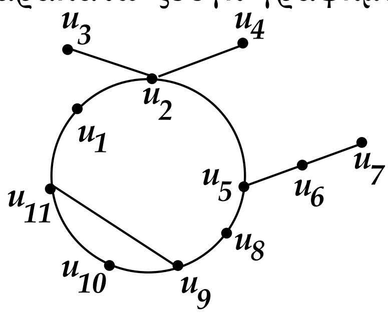
vii) Το βαρύκεντρο του G.

(2) Να εξετασθεί αν είναι ισόμορφα τα παρακάτω ζεύγη γραφημάτων.

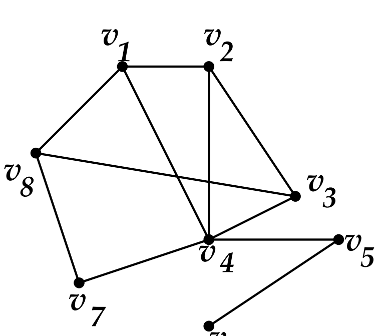
*G*1



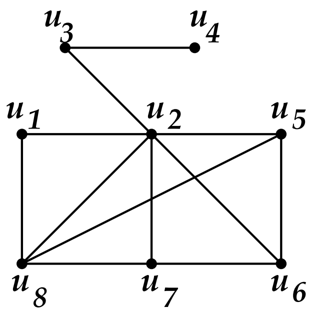
*G*2



*H*1

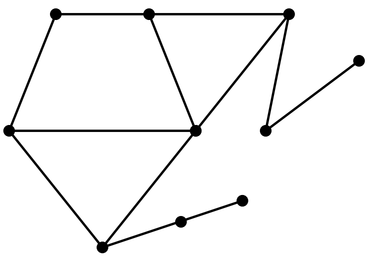


*H*2

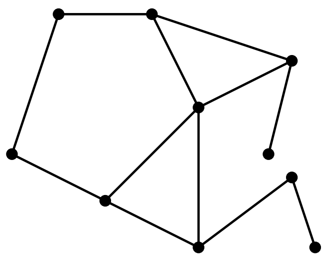


(3) Να δειχθεί ότι δεν είναι ισόμορφα τα παρακάτω γραφήματα:

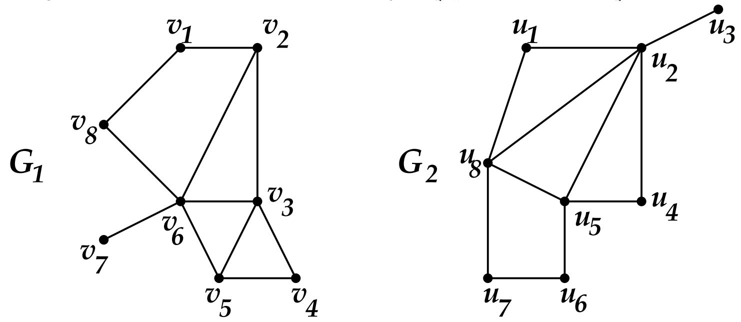
H

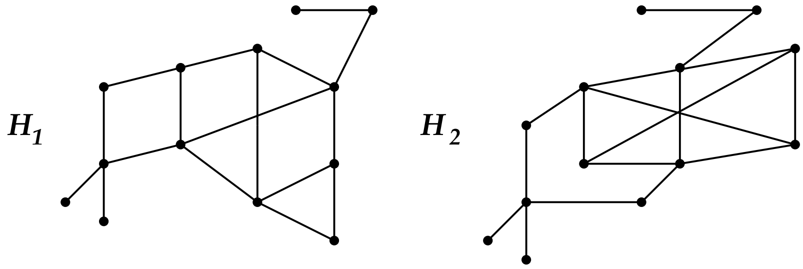


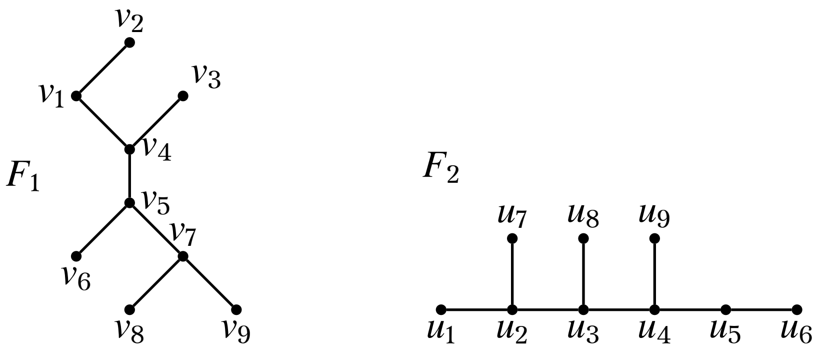
H′



(4) Να εξετασθεί αν είναι ισόμορφα τα παρακάτω ζεύγη γραφημάτων.



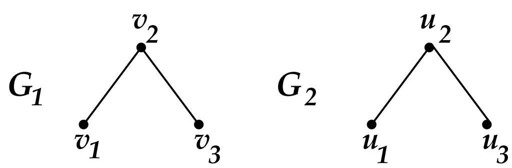




(5) Για τα γραφήματα

i)

ii)

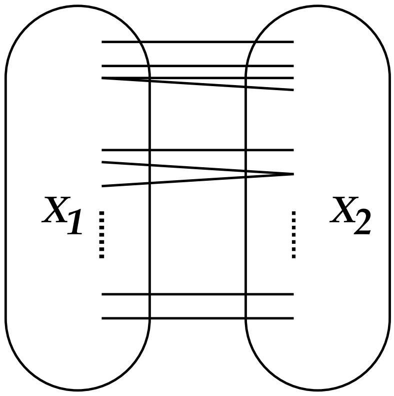


να ορισθούν τα $G_1 \cup G_2$, $G_1 + G_2$, $G_1 \times G_2$, $G_1(G_2)$.

10. ΔΙΜΕΡΗ ΓΡΑΦΗΜΑΤΑ

Ένα γράφημα G λέγεται **διμερές** (bipartite) αν το σύνολο X των κορυφών του μπορεί να διαμεριστεί σε δύο υποσύνολα X_1, X_2 τέτοια ώστε κάθε $e \in E$ ενώνει μια κορυφή του X_1 με μια κορυφή του X_2. Η διαμέριση X_1, X_2 ονομάζεται **διμερής διαμέριση** (bipartition) των κορυφών του G.

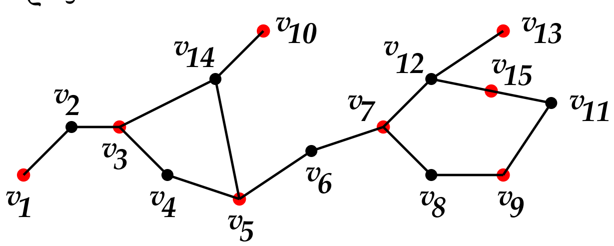
G



Παραδείγματα

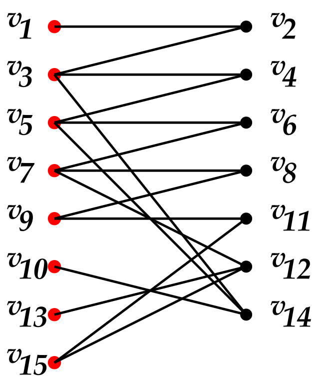
Το γράφημα G είναι διμερές:

G

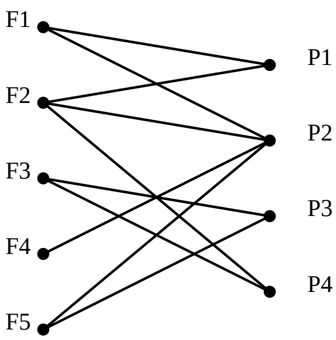


Πράγματι, το G γράφεται:

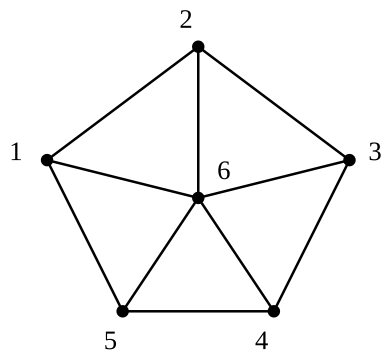
G



Το γράφημα G_1 είναι διμερές, ενώ το γράφημα G_2 δεν είναι διμερές.



G_1



G_2

Πρόταση 25. *Ένα γράφημα είναι διμερές αν και μόνο αν όλοι οι κύκλοι του είναι άρτιου μήκους. (Ή, ισοδύναμα δεν έχει κύκλους περιττού μήκους.)*

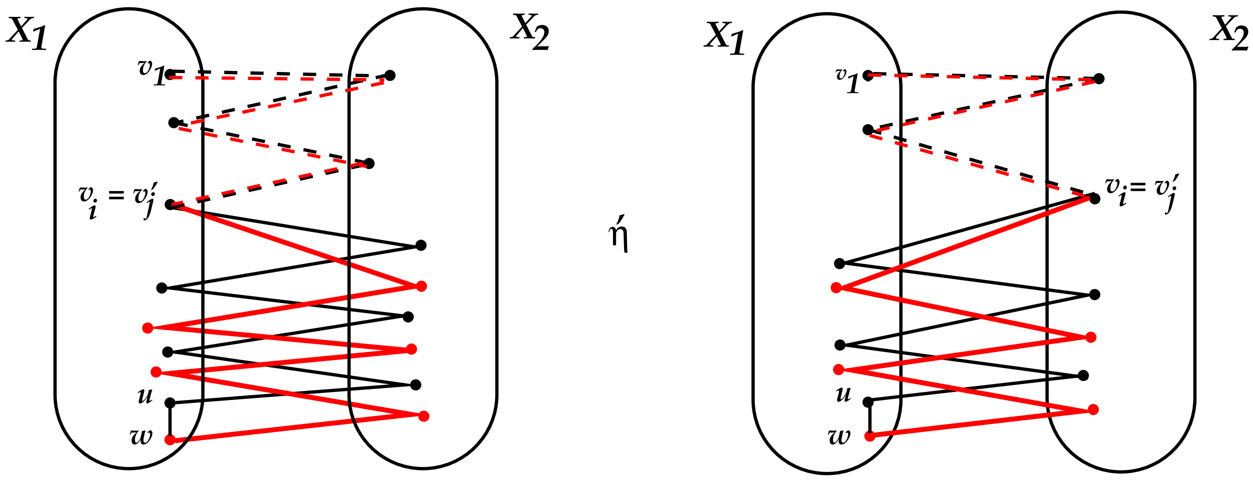
Απόδειξη. *Ευθύ*: Έστω G διμερές, με $X = X_1 \cup X_2$, $(X_1 \cap X_2 = \emptyset)$ και $(v_1, v_2, \ldots, v_n, v_1)$ ένας κύκλος του με $v_1 \in X_1$. Αφού σε κάθε «βήμα» πηγαίνουμε από το ένα σύνολο κόμβων στο άλλο και αφού καταλήγουμε στο ίδιο σύνολο X_1 από το οποίο ξεκινήσαμε θα έχουμε κάνει άρτιο αριθμό «βημάτων», δηλαδή ο κύκλος θα έχει άρτιο μήκος.

Αντίστροφο: Έστω G συνεκτικό (διαφορετικά, εργαζόμαστε αντίστοιχα σε κάθε συνιστώσα του). Έστω $v_1 \in X$. Διαμερίζουμε το X σε X_1, X_2 ως εξής: Το X_1 αποτελείται από το v_1 και όλους τους κόμβους του X που απέχουν άρτια απόσταση από το v_1, ενώ $X_2 = X \setminus X_1$. Θα δείξουμε ότι κάθε $\{u, w\}$ του $E(G)$ ενώνει ένα κόμβο του X_1 με ένα κόμβο του X_2 (οπότε το G είναι διμερές).

Πράγματι, αν ο $\{u, w\}$ ένωνε δύο κόμβους του X_1 (δες επόμενο σχήμα) ή, όμοια, του X_2, τότε έστω

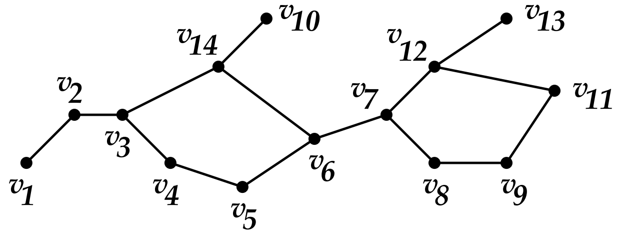
$$(v_1, v_2, \ldots, v_{k-1}, v_k = u) \text{ και } (v_1, v'_2, \ldots, v'_{\lambda-1}, v'_\lambda = w)$$

τα συντομότερα $v_1 - u$ και $v_1 - w$ μονοπάτια, τα οποία θα ήταν άρτιου μήκους (αφού αρχίζουν και τελειώνουν στο ίδιο σύνολο X_1). Άρα k, λ : περιττοί. Έστω $v_i = v'_j$ ο τελευταίος κοινός κόμβος των μονοπατιών αυτών.



Τότε ο κύκλος $(v_i, v_{i+1}, \ldots, v_k = u, w = v'_\lambda, \ldots, v'_{j+1}, v'_j)$ έχει μήκος : $(k - i) + 1 + (\lambda - j) = (k + \lambda) - (i + j) + 1$. Αλλά οι i, j είναι και οι δύο περιττοί (αν $v_i = v'_j \in X_1$), ή και οι δύο άρτιοι (αν $v_i = v'_j \in X_2$). Σε κάθε περίπτωση λοιπόν, $i + j$: άρτιος. Επίσης $k + \lambda$: άρτιος (αφού k, λ : περιττοί). Άρα το μήκος του παραπάνω κύκλου είναι περιττό. Άτοπο. □

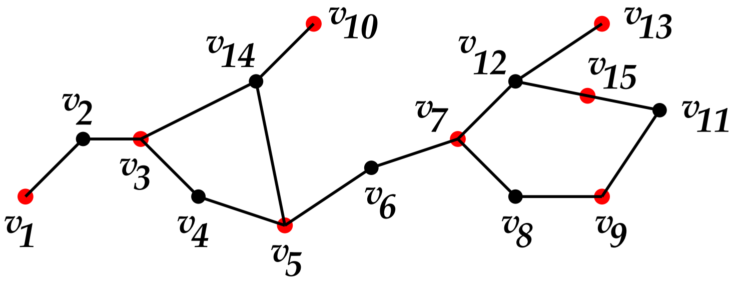
Παρατηρήσεις: Από την προηγούμενη πρόταση προκύπτει ένα κριτήριο για το εάν γράφημα είναι διμερές. Για παράδειγμα, το γράφημα



δεν είναι διμερές διότι περιέχει κύκλο περιττού μήκους: (v_7, v_{12}, v_{11}, v_9, v_8, v_7)

Επίσης, η απόδειξη δίνει έναν αλγόριθμο για την εύρεση της διμερούς διαμέρισης X_1, X_2 των κορυφών ενός (συνεκτικού) διμερούς γραφήματος: Επιλέγουμε αυθαίρετα μια κορυφή v και ορίζουμε X_1 (αντ. X_2) το σύνολο των κορυφών που απέχουν άρτια (αντ. περιττή) απόσταση από την v. (Αν το γράφημα δεν είναι συνεκτικό επαναλαμβάνουμε την ίδια διαδικασία σε κάθε συνεκτική συνιστώσα.)

Για το συνεκτικό διμερές γράφημα



η διμερής διαμέριση X_1, X_2 προκύπτει διαμερίζοντας τις κορυφές του με βάση την απόσταση τους από την κορυφή v_1, οπότε έχουμε

$$X_1 = \{v_1, v_3, v_5, v_7, v_9, v_{10}, v_{13}, v_{15}\}.$$
$$X_2 = \{v_2, v_4, v_6, v_8, v_{11}, v_{12}, v_{14}\}.$$

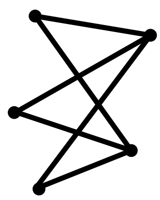
Τέλος, από την προηγούμενη απόδειξη, προκύπτει ότι στην περίπτωση που ένα διμερές γράφημα G είναι συνεκτικό υπάρχει μοναδική διμερής διαμέριση των κορυφών του σε δύο σύνολα X_1, X_2. Αν το G είναι μη συνεκτικό, με τουλάχιστον ένα δεσμό, και αποτελείται από k συνεκτικές συνιστώσες τότε εύκολα προκύπτει ότι υπάρχουν 2^{k-1} διαφορετικές διμερείς διαμερίσεις X_1, X_2 των κορυφών του.

ΠΛΗΡΕΣ ΔΙΜΕΡΕΣ ΓΡΑΦΗΜΑ

Χρησιμοποιήσαμε ήδη το συμβολισμό K_n για το πλήρες γράφημα με n κόμβους. Με $K_{n,m}$ συμβολίζουμε ένα διμερές γράφημα $G = (X, E)$ με $X = X_1 \cup X_2$, $X_1 \cap X_2 = \emptyset$, $|X_1| = n$, $|X_2| = m$ και τέτοιο ώστε για κάθε $v \in X_1$ και για κάθε $u \in X_2$ να ισχύει ότι $\{v, u\} \in E$. Το $K_{n,m}$ ονομάζεται **πλήρες διμερές γράφημα** (complete bipartite graph).

Παράδειγμα :

$K_{3,2}$



Πρόταση 26. *Για κάθε* $m, n \in \mathbb{N}^*$ *ισχύει ότι* $|E(K_{n,m})| = n \cdot m$.

Απόδειξη. Κάθε δεσμός του $K_{n,m}$ έχει ακριβώς ένα άκρο του στο σύνολο X_1 όπου $|X_1| = n$. Επομένως, μπορούμε να μετρήσουμε τους δεσμούς με βάση τα άκρα τους στο X_1. Κάθε κορυφή $v \in X_1$ είναι άκρο σε m δεσμούς, άρα συνολικά υπάρχουν $n \cdot m$ δεσμοί. □

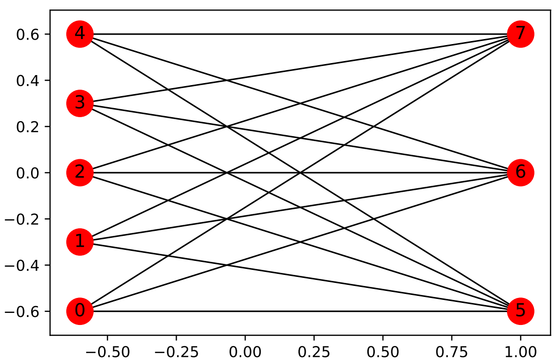
Η ιδέα της προηγούμενης απόδειξης μπορεί να χρησιμοποιηθεί για την λύση της άσκησης 1.

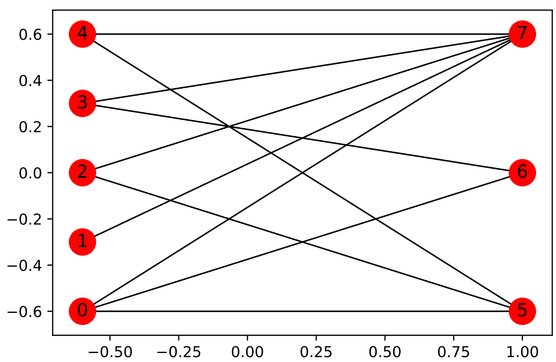
Παρατήρηση: Το γράφημα $K_{n,n}$ έχει $2n$ κορυφές και n^2 δεσμούς και επειδή είναι διμερές δεν περιέχει τρίγωνα (κύκλους μήκους 3). Μάλιστα έχει την ιδιότητα ότι περιέχει τον μέγιστο δυνατό αριθμό δεσμών μεταξύ όλων των γραφημάτων που έχουν $2n$ κορυφές και δεν περιέχουν τρίγωνα.

Μπορούμε να κατασκευάζουμε διμερή γραφήματα χρησιμοποιώντας τις μεθόδους `complete_bipartite_graph(n,m)` και `bipartite.gnmk_random_graph(n,m,k)` που κατασκευάζουν το πλήρες διμερές γράφημα $K_{m,n}$ και ένα τυχαίο διμερές γράφημα με $|X_1| = n$, $|X_2| = m$ και k δεσμούς, όπου $k \leq n \cdot m$.

```
import networkx as nx
import matplotlib.pyplot as plt

#create the complete bipartite graph K_{n,m}
n, m = 5, 3 #number of nodes in each set
B = nx.bipartite.complete_bipartite_graph(n,m)
#draw the graph using the bipartite layout algorithm
pos = nx.bipartite_layout(B,range(n))
nx.draw_networkx(B,pos)
plt.show()

#create a random bipartite graph with
#n,m nodes in each set resp. and k <= m*n edges
n, m = 5, 3 #number of nodes in each set
k = 10 #number of edges
B2 = nx.bipartite.gnmk_random_graph(n,m,k)
pos = nx.bipartite_layout(B2,range(n))
nx.draw_networkx(B2,pos)
plt.show()
```

Output:





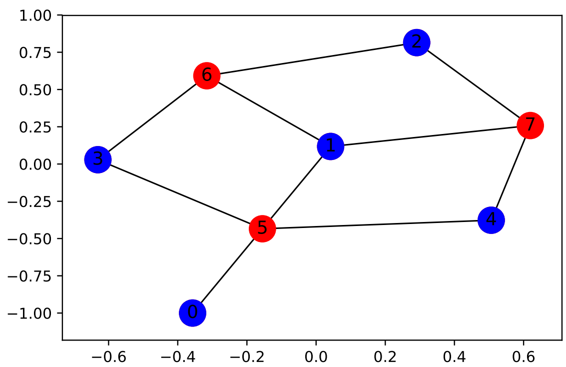
Μπορούμε να ελέγξουμε αν ένα γράφημα G είναι διμερές χρησιμοποιώντας την μέθοδο `is_bipartite(G)`. Στην περίπτωση όπου το G είναι **συνεκτικό** διμερές γράφημα με την μέθοδο `bipartite.sets(G)` μπορούμε να βρούμε την διμερή διαμέριση X_1, X_2. (Αν το G δεν είναι συνεκτικό, η μέθοδος `bipartite.sets(G)` δεν λειτουργεί και πρέπει να δουλέψουμε ξεχωριστά σε κάθε συνεκτική συνιστώσα)

```
import networkx as nx
import matplotlib.pyplot as plt
```

```
n, m, k = 5, 3, 10
G = nx.bipartite.gnmk_random_graph(n,m,k)
pos = nx.layout.kamada_kawai_layout(G)
nx.draw_networkx(G,pos)
if nx.is_bipartite(G):
    X1 = []
    X2 = []
    Gcc = nx.connected_components(G)
    for cc in Gcc:
        G1 = G.subgraph(cc)
        X, Y = nx.bipartite.sets(G1)
        X1.extend(X)
        X2.extend(Y)
    print("The graph is bipartite")
    print("X1:",X1, "X2:",X2)
    G1 = G.subgraph(X1)
    nx.draw_networkx_nodes(G1,pos,node_color='blue',width=3.0)
else:
    print("The graph is not bipartite")
plt.show()
```

Output:

```
The graph is bipartite
X1: [0, 1, 2, 3, 4] X2: [5, 6, 7]
```



Παρακάτω δίδεται μια εναλλακτική υλοποίηση της μεθόδου `bipartite.sets(G)` η οποία χρησιμοποιεί την αρτιότητα της απόστασης από μια τυχαία κορυφή (σε κάθε συνεκτική συνιστώσα) για τον υπολογισμό της διμερούς διαμέρισης των κορυφών ενός διμερούς γραφήματος:

```
import networkx as nx
import matplotlib.pyplot as plt
from random import choice

n, m, k = 5, 3, 10
G = nx.bipartite.gnmk_random_graph(n,m,k)
pos = nx.layout.kamada_kawai_layout(G)
nx.draw_networkx(G,pos)
if nx.is_bipartite(G):
    nx.bipartite.sets
    X1 = []
    X2 = []
    Gcc = nx.connected_components(G)
```

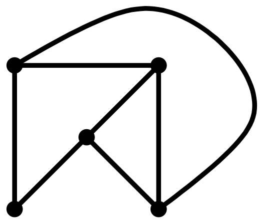
```
    for cc in Gcc:
        G1 = G.subgraph(cc)
        random_node = choice(list(G1.nodes()))
        distances = nx.shortest_path_length(G1,random_node)
        for v in G1:
            if(distances[v] % 2 == 0):
                X1.append(v)
            else:
                X2.append(v)
    print("The graph is bipartite")
    print("X1:",X1, "X2:",X2)
    G1 = G.subgraph(X1)
    nx.draw_networkx_nodes(G1,pos,node_color='blue',width=3.0)
else:
    print("The graph is not bipartite")
plt.show()
```

11. ΕΠΙΠΕΔΑ ΓΡΑΦΗΜΑΤΑ

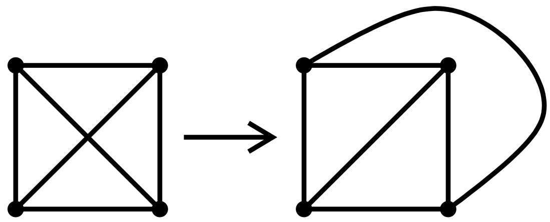
Επίπεδο (planar) λέγεται ένα γράφημα που μπορεί να απεικονισθεί στο επίπεδο, έτσι ώστε :

α) Οι κόμβοι του να είναι διακεκριμένα σημεία.
β) Οι δεσμοί του να είναι απλές, επίπεδες καμπύλες.
γ) Κάθε ζεύγος δεσμών (αν συναντιούνται), συναντιούνται μόνο στους κόμβους.

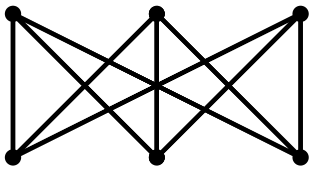
Παράδειγμα :



Επίπεδο γράφημα



Επίπεδο γράφημα



Μη επίπεδο γράφημα

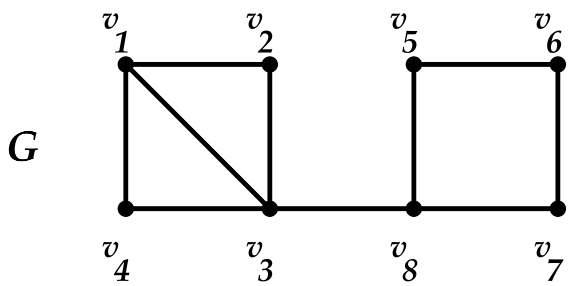
Επίπεδο τοπολογικό γράφημα (plane graph) λέγεται ένα επίπεδο γράφημα που έχει ήδη απεικονισθεί στο επίπεδο, έτσι ώστε να ικανοποιεί τις συνθήκες α), β) και γ).

Παράδειγμα : Το πρώτο από τα παραπάνω γραφήματα είναι επίπεδο τοπολογικό γράφημα.

Παρατήρηση : Στο "**πρόβλημα σύνδεσης**" το ερώτημα που τίθεται (κατά πόσον κάποιοι κόμβοι μπορούν να συνδεθούν με κάποιους άλλους χωρίς να υπάρχουν "διασταυρώσεις") είναι ουσιαστικά το ερώτημα : κατά πόσον το προκύπτον γράφημα είναι επίπεδο. (Εφαρμογές : Πληροφορική, ηλεκτρολογία (συνδεσμολογίες), συγκοινωνίες κ.λπ.). Ένα θεώρημα σχετικό με το θέμα αυτό (ικανή και αναγκαία συνθήκη για να είναι ένα γράφημα μη επίπεδο) είναι το διάσημο θεώρημα του Kuratowski, που θα δούμε αργότερα.

Ορίζουμε σαν **έδρες** (faces) ενός επίπεδου τοπολογικού γραφήματος G τις κλειστές περιοχές του επίπεδου που ορίζονται από τους δεσμούς του γραφήματος. Η ανοιχτή περιοχή ονομάζεται **εξωτερική έδρα** (outer face) του G.

Παράδειγμα :



Το γράφημα G έχει τρεις έδρες πλην της εξωτερικής : Τα "τρίγωνα" που ορίζονται από τους κύκλους (v_1, v_2, v_3, v_1) και (v_1, v_3, v_4, v_1) και το "τετράγωνο" που ορίζεται από τον κύκλο $(v_5, v_6, v_7, v_8, v_5)$.

Πρόταση 27 (**Τύπος του Euler**). *Έστω $G = (V, E)$ ένα συνεκτικό επίπεδο τοπολογικό γράφημα με $|F| = |F(G)|$ έδρες (συμπεριλαμβανόμενης και της εξωτερικής). Τότε*

$$|V| - |E| + |F| = 2.$$

Παράδειγμα : Στο προηγούμενο γράφημα G έχουμε :
$|V| = 8$, $|E| = 10$, $|F| = 4$ και πράγματι $8 - 10 + 4 = 2$.
Παρατηρήσεις :

(1) Από τον τύπο του Euler προκύπτει ότι αν ένα γράφημα είναι επίπεδο κάθε αναπαράσταση του ως επίπεδο τοπολογικό γράφημα θα έχει πάντα τον ίδιο αριθμό εδρών.
(2) Ο τύπος του Euler ισχύει και για μη απλά γραφήματα, δηλαδή για γραφήματα που περιέχουν βρόχους ή και πολλαπλούς δεσμούς ανάμεσα στις κορυφές τους.
(3) Ο τύπος του Euler ισχύει και για γραφήματα τα οποία έχουν απεικονισθεί πάνω σε μια σφαίρα, έτσι ώστε να ικανοποιούν τις συνθήκες α), β) και γ). Σε αυτή την περίπτωση ιδιαίτερο ενδιαφέρον έχουν τα πολύεδρα με κορυφές πάνω στην επιφάνεια της σφαίρας.

Λήμμα 28.

(1) Σε κάθε επίπεδο γράφημα $G = (V, E)$, με $|E| \geq 2$, ισχύει ότι

$$3|F| \leq 2|E|$$

(2) Σε κάθε επίπεδο διμερές γράφημα $G = (V, E)$ με $|E| \geq 2$ ισχύει ότι

$$2|F| \leq |E|$$

Απόδειξη. Έστω $s(G)$ ο αριθμός των ζευγών (e, f) του G για τα οποία ο δεσμός e συνορεύει με την έδρα f.

Μπορούμε να μετρήσουμε τα ζεύγη (e, f) με δύο τρόπους: Αθροίζοντας για κάθε έδρα f το πλήθος των δεσμών e που συνορεύουν με αυτήν, ή αθροίζοντας για κάθε δεσμό e το πλήθος των εδρών f οι οποίες συνορεύουν με αυτόν.

(1) Με τον πρώτο τρόπο, κάθε έδρα συνορεύει με τουλάχιστον 3 δεσμούς, επομένως

$$s(G) \geq 3|F|$$

Με τον δεύτερο τρόπο, κάθε δεσμός συνορεύει το πολύ με δύο έδρες, οπότε

$$s(G) \leq 2|E|$$

Άρα

$$3|F| \leq 2|E|.$$

(2) Επειδή το γράφημα είναι διμερές, δεν περιέχει κύκλους περιττού μήκους, άρα, με τον πρώτο τρόπο, κάθε έδρα συνορεύει με τουλάχιστον 4 δεσμούς, επομένως

$$s(G) \geq 4|F|$$

Με τον πρώτο τρόπο, κάθε δεσμός συνορεύει το πολύ με δύο έδρες, οπότε

$$s(G) \leq 2|E|$$

Άρα

$$4|F| \leq 2|E| \Leftrightarrow 2|F| \leq |E|. \qquad \square$$

Πόρισμα 29 (**Ανισότητα κορυφών-δεσμών σε επίπεδα γραφήματα**).

(1) Σε κάθε επίπεδο γράφημα $G = (V, E)$ ισχύει ότι

$$|E| \leq 3|V| - 6$$

(2) Σε κάθε επίπεδο διμερές γράφημα $G = (V, E)$ ισχύει ότι

$$|E| \leq 2|V| - 4$$

Απόδειξη.

(1) Από το προηγούμενο λήμμα για κάθε επίπεδο γράφημα $G = (V, E)$ ισχύει ότι

$$3|F| \leq 2|E|$$

Όμως, από τον τύπο του Euler έχουμε ότι $|F| = |E| - |V| + 2$ οπότε

$$3(|E| - |V| + 2) \leq 2|E| \Leftrightarrow |E| \leq 3|V| - 6$$

(2) Άσκηση. $\square$

Άσκηση 5. *Τα γραφήματα K_5 και $K_{3,3}$ δεν είναι επίπεδα.*

Λύση. Για το γράφημα K_5 έχουμε ότι $|V(K_5)| = 5$ και $|E(K_5)| = \binom{5}{2} = 10$.
Από το προηγούμενο πόρισμα, έχουμε ότι αν το K_5 είναι επίπεδο πρέπει

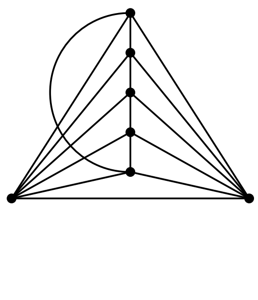
$$|E(K_5)| \leq 3|V(K_5)| - 6 \Leftrightarrow 10 \leq 3 \cdot 5 - 6 \Leftrightarrow 10 \leq 9$$

άτοπο, άρα το K_5 δεν είναι επίπεδο.
Για το γράφημα $K_{3,3}$ έχουμε ότι $|V(K_{3,3})| = 6$ και $|E(K_{3,3})| = 3 \cdot 3 = 9$.
Επειδή το $K_{3,3}$ είναι διμερές, από το προηγούμενο πόρισμα, έχουμε ότι αν το $K_{3,3}$ είναι επίπεδο πρέπει

$$|E(K_{3,3})| \leq 2|V(K_{3,3})| - 4 \Leftrightarrow 9 \leq 2 \cdot 6 - 4 \Leftrightarrow 9 \leq 8$$

άτοπο, άρα το $K_{3,3}$ δεν είναι επίπεδο. $\square$

Άσκηση 6. *Να εξετασθεί αν το παρακάτω γράφημα είναι επίπεδο ή όχι.*



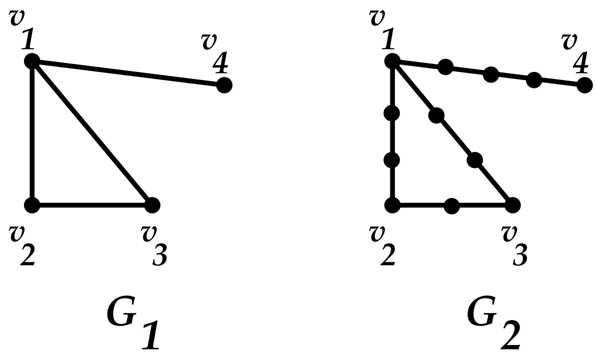
 # κορυφών = 7,
 # δεσμών = 16

Παρατήρηση: Προφανώς, αν ένα γράφημα G έχει ως υπογράφημα ένα μη επίπεδο γράφημα H, τότε το G είναι μη επίπεδο.

ΤΟ ΘΕΩΡΗΜΑ ΤΟΥ KURATOWSKI

Εάν ένα γράφημα G_2 προκύπτει από ένα γράφημα G_1 με αντικατάσταση ενός ή περισσότερων δεσμών του G_1 από ένα μονοπάτι μεταξύ των αντίστοιχων κόμβων (δηλαδή, με την παρεμβολή νέων κόμβων), τότε το G_2 λέγεται **εκλέπτυνση** ή **υποδιαίρεση** του G_1.

Παράδειγμα :



Το G_2 είναι μια εκλέπτυνση του G_1.

Πρόταση 30 (**Θεώρημα Kuratowski**). *Ένα γράφημα είναι μη επίπεδο αν και μόνο αν έχει ένα υπογράφημα που είναι εκλέπτυνση του K_5 ή του $K_{3,3}$.*

Σχετικές ασκήσεις: 8, 9

Παρατήρηση: Το Θεώρημα του Kuratoswki συνδέει μια τοπολογική ιδιότητα (την αναπαράσταση ενός γραφήματος στο επίπεδο χωρίς τεμνόμενους δεσμούς) με μια συνδυαστική ιδιότητα (την μη εμφάνιση ως υπογραφημάτων των εκλεπτύνσεων του K_5 και του $K_{3,3}$). Επειδή, ο έλεγχος αυτός μπορεί να γίνει σε πολυωνυμικό χρόνο έπεται ότι μπορούμε να ελέγξουμε αν ένα γράφημα είναι ή όχι επίπεδο σε πολυωνυμικό χρόνο.

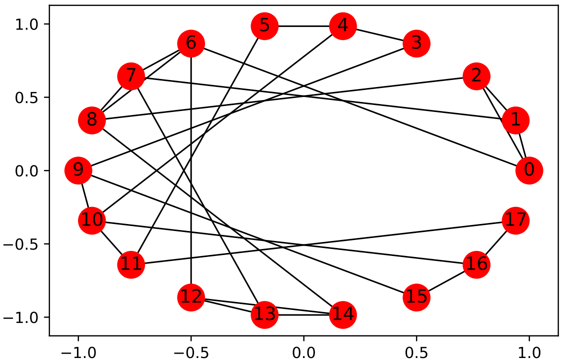
Χρησιμοποιώντας την μέθοδο `check_planarity(G)` της βιβλιοθήκης `networkx` μπορούμε αφενός να ελέγξουμε αν ένα γράφημα είναι επίπεδο και αφετέρου, εφόσον είναι επίπεδο, να το σχεδιάσουμε ως επίπεδο τοπολογικό γράφημα:

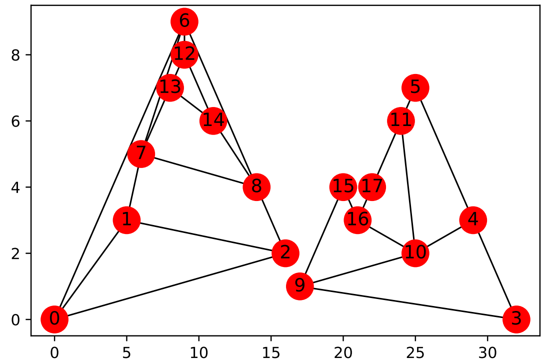
```
import networkx as nx
import matplotlib.pyplot as plt

K5 = nx.complete_graph(5)
#check_planarity(G) returns the tuple: bool is_planar, PlanarEmbedding
#is_planar is True iff G is planar
#PlanarEmbedding is a combinatorial description of a planar embedding
#to be used with nx.combinatorial_embedding_to_pos(embedding)
#to get the x,y coordinates of each vertex
if nx.check_planarity(K5)[0]:
    print("The graph is planar")
else:
    print("The graph is not planar")

```

```python
#Create a planar graph G using some operations
G1 = nx.cycle_graph(3)
G2 = nx.path_graph(3)
G = nx.disjoint_union(G1,G2)
G = nx.cartesian_product(G2, G)
G = nx.disjoint_union(G, nx.Graph())
#A non-planar drawing
pos = nx.layout.shell_layout(G)
nx.draw_networkx(G,pos)
plt.show()
is_planar, embedding = nx.check_planarity(G)
#A planar drawing of G (if it exists)
if is_planar:
    pos = nx.combinatorial_embedding_to_pos(embedding)
    nx.draw_networkx(G,pos)
    plt.show()
```

Output:

```
The graph is not planar
```





Στην περίπτωση που ένα γράφημα είναι μη επίπεδο, η εύρεση μιας εκλέπτυνσης του K_5 ή του $K_{3,3}$ μπορεί να επιτευχθεί με την επαναληπτική μέθοδο `get_counterexample(G)`, η οποία χρησιμοποιεί την μέθοδο `check_planarity(G)`:

Η ιδέα της μεθόδου είναι ότι σε ένα μη επίπεδο γράφημα υπάρχουν δεσμοί του γραφήματος των οποίων η αφαίρεση καταστρέφει την ύπαρξη των εκλεπτύνσεων του K_5 και $K_{3,3}$, οπότε το γράφημα που προκύπτει είναι επίπεδο. (Ο λόγος που συμβαίνει αυτό είναι ότι κάθε γνήσιο υπογράφημα μιας εκλέπτυνσης των K_5 και $K_{3,3}$ είναι επίπεδο γράφημα). Επομένως, οι δεσμοί αυτοί αποτελούν μέρος αυτών των εκλεπτύνσεων.

Προκειμένου να τους βρούμε, αφαιρούμε από το G έναν-έναν τους δεσμούς του $e_1, e_2, \ldots$, οπότε προκύπτει μια ακολουθία G_i από μη επίπεδα υπογραφήματα του G όπου $G_0 = G$ και G_{i+1} είναι ίσο με

- $G_i - e_i$, αν το γράφημα $G_i - e_i$ είναι μη επίπεδο. (Δηλαδή, αφαιρούμε τον e_i οριστικά, διότι υπάρχει εκλέπτυνση των K_5 ή/και $K_{3,3}$ στην οποία δεν συμμετέχει ο δεσμός e_i.)
- G_i, αν το γράφημα $G_i - e_i$ είναι επίπεδο. (Στην περίπτωση αυτή, ο δεσμός e_i ανήκει οπωσδήποτε στην μοναδική εκλέπτυνση των K_5 ή $K_{3,3}$ που περιέχει το γράφημα G_i.)

Στο τέλος, οι δεσμοί e που δεν αφαιρέθηκαν αποτελούν τους δεσμούς της ζητούμενης εκλέπτυνσης του K_5 ή του $K_{3,3}$.

```
def get_counterexample(G):
    # copy graph
    G = nx.Graph(G)

    if nx.check_planarity(G)[0]:
        raise nx.NetworkXException("G is planar - no counter example.")

    # find Kuratowski subgraph
    subgraph = nx.Graph()
    for u in G:
        nbrs = list(G[u]) #get the neighbors of u
        for v in nbrs:
            G.remove_edge(u, v)
            if nx.check_planarity(G)[0]:
                G.add_edge(u, v)          #put back the edge {u,v}
                subgraph.add_edge(u, v)   #edge {u,v} belongs to the Kuratowski graph

    return subgraph
```

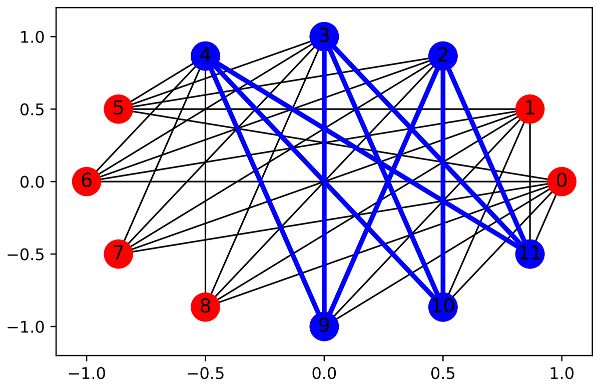
Ακολουθούν 3 παραδείγματα μη επίπεδων γραφημάτων στα οποία βρίσκουμε τις αντίστοιχες εκλεπτύνσεις του K_5 ή του $K_{3,3}$ με την βοήθεια της μεθόδου `get_counterexample(G)`:

```
import networkx as nx
import matplotlib.pyplot as plt

def find_and_draw_kuratowski_in_nonplanar_graph(G,name):
    #pos = nx.nx_agraph.graphviz_layout(G)
    pos = nx.layout.shell_layout(G)
    nx.draw_networkx(G,pos)
    kuratowski = nx.planarity.get_counterexample(G)
    print("The Kuratowski graph consists of nodes:",kuratowski.nodes())
    nx.draw_networkx(kuratowski,pos,node_color="blue",edge_color="blue",width=3.0)
    plt.savefig(name)
    plt.show()

#1st example of finding Kuratowski subgraph in non-planar graph
K57 = nx.bipartite.complete_bipartite_graph(5,7)
find_and_draw_kuratowski_in_nonplanar_graph(K57,"k57.eps")

#2nd example of finding Kuratowski subgraph in non-planar graph
n, k, p = 20, 3, 0.5
```

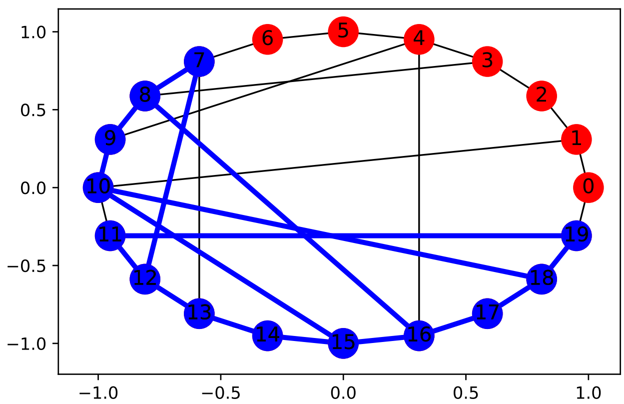
```
G = nx.random_graphs.newman_watts_strogatz_graph(n,k,p)
find_and_draw_kuratowski_in_nonplanar_graph(G,"k2.eps")

#3rd example of finding Kuratowski subgraph in non-planar graph
G1 = nx.cycle_graph(6)
G2 = nx.power(nx.cycle_graph(5),2)
G = nx.cartesian_product(G2, G1)
find_and_draw_kuratowski_in_nonplanar_graph(G,"k1.eps")
```

Output:

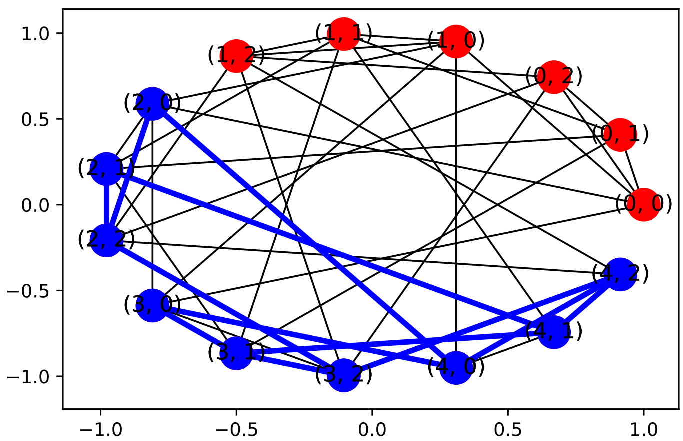
```
The Kuratowski graph consists of nodes:
    [2, 9, 10, 11, 3, 4]
```




```
The Kuratowski graph consists of nodes:
    [7, 8, 12, 9, 16, 10, 15, 18, 11, 19,
    13, 14, 17]
```




```
The Kuratowski graph consists of nodes:
    [(2, 0), (4, 0), (2, 2), (2, 1), (4,
    1), (3, 2), (3, 0), (3, 1), (4, 2)]
```



Σχετικός με το πρόβλημα σύνδεσης είναι και ο παρακάτω ορισμός :

Αριθμός διασταυρώσεων (crossing number) $cr(G)$ ενός γραφήματος G είναι ο ελάχιστος αριθμός διασταυρώσεων των δεσμών του G ανά δύο, όταν το G ζωγραφιστεί στο επίπεδο.

Προφανώς $cr(G) = 0$ αν και μόνο αν το G είναι επίπεδο γράφημα. Έχει αποδειχθεί ότι ο αριθμός διασταυρώσεων ενός πλήρους γραφήματος K_n ικανοποιεί την ανισότητα :

$$cr(K_n) \leq \frac{1}{4}\lfloor\frac{n}{2}\rfloor\lfloor\frac{n-1}{2}\rfloor\lfloor\frac{n-2}{2}\rfloor\lfloor\frac{n-3}{2}\rfloor$$

(και μάλιστα, για $n \leq 12$, ισχύει το "=").

Επίσης, έχει αποδειχθεί ότι

$$cr(K_{m,n}) \leq \lfloor\frac{m}{2}\rfloor\lfloor\frac{m-1}{2}\rfloor\lfloor\frac{n}{2}\rfloor\lfloor\frac{n-1}{2}\rfloor$$

(και μάλιστα, όταν $m \leq n$ και είτε $m \leq 6$ είτε $m = 7$ και $n \leq 10$, ισχύει το "=").

Πρόταση 31. *Έστω $G = (V, E)$ ένα γράφημα δεσμών τότε*

$$|E(G)| - 3|V(G)| + 6 \leq cr(G)$$

Απόδειξη. Διακρίνουμε δύο περιπτώσεις:

Αν το G είναι επίπεδο, τότε $cr(G) = 0$ και $|E(G)|-3|V(G)|+6 \leq 0$, άρα η ανισότητα ισχύει.

Αν το G δεν είναι επίπεδο, τότε κάθε αναπαράσταση του στο επίπεδο θα περιέχει τουλάχιστον ένα σημείο διασταύρωσης. Έστω μια αναπαράσταση του G με $cr(G) = c$ σημεία διασταύρωσης.

Αν στο γράφημα G προσθέσουμε c επιπλέον κορυφές στα σημεία διασταύρωσης των δεσμών του, τότε θα προκύψει ένα γράφημα G' το οποίο είναι επίπεδο και έχει $|V(G)| + c$ κορυφές και $|E(G)| + 2c$ δεσμούς.

Επομένως, από το Πόρισμα 29 για το γράφημα G' θα ισχύει ότι

$$|E(G')| \leq 3|V(G')| - 6 \Leftrightarrow$$

$$|E(G)| + 2c \leq 3(|V(G)| + c) - 6 \Leftrightarrow$$

$$|E(G)| - 3|V(G)| + 6 \leq c$$

□

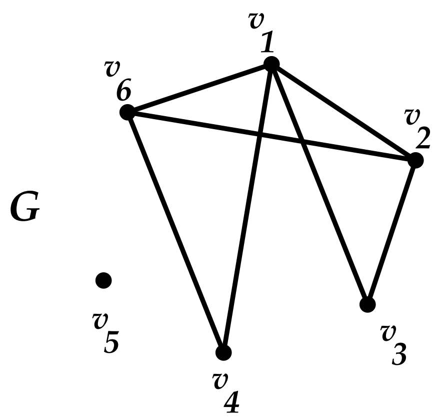
12. ΜΗΤΡΑ ΓΡΑΦΗΜΑΤΟΣ ΔΕΣΜΩΝ

Έστω $G = (V, E)$. Ορίζουμε την $|V| \times |V|$ μήτρα (πίνακα) M_G ή M του G ως εξής:

$$M = [m_{ij}], \text{ με } m_{ij} = \begin{cases} 1, & \text{αν } \{v_i, v_j\} \in E \\ 0, & \text{αν } \{v_i, v_j\} \notin E. \end{cases}$$

Η μήτρα αυτή ονομάζεται **μήτρα (γειτονικότητας) του γραφήματος δεσμών**.

Παράδειγμα: Στο γράφημα G



αντιστοιχεί η μήτρα

$$M = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

Παρατηρήσεις:

1. Η μήτρα M είναι προφανώς συμμετρική.
2. Ισχύει ότι

$$\sum_{j=1}^{|V|} m_{ij} = \sum_{j=1}^{|V|} m_{ji} = d(v_i)$$

(δηλαδή το άθροισμα των στοιχείων της i γραμμής ισούται με το άθροισμα των στοιχείων της i στήλης και με τον βαθμό του κόμβου v_i).

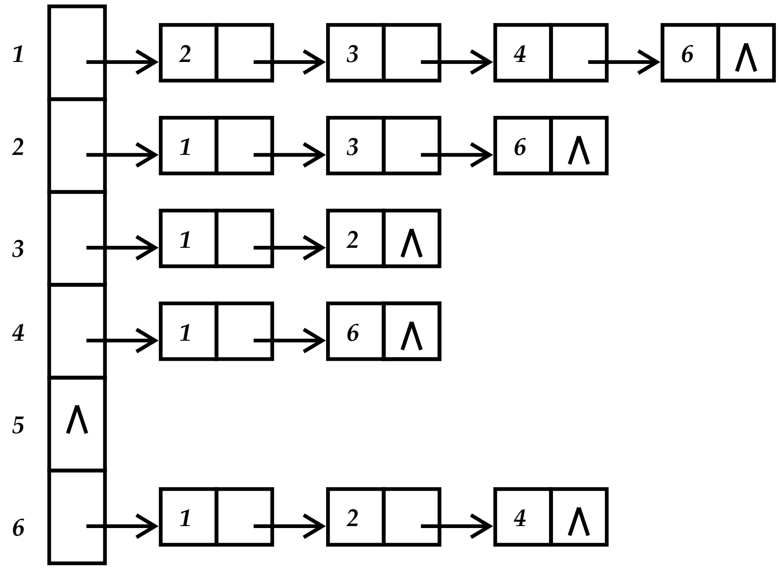
Για να εκφραστούν και να υλοποιηθούν σαφέστερα κάποιοι αλγόριθμοι στον υπολογιστή, χρησιμοποιούνται και οι **λίστες γειτονικότητας**.

Σ' αυτές, κάθε γραμμή (λίστα) αντιστοιχεί σε ένα κόμβο v_i, ο δείκτης i του οποίου εμφανίζεται ως επικεφαλής δείκτης της λίστας. Τα υπόλοιπα στοιχεία της λίστας έχουν τη μορφή

κόμβος	δείκτης

όπου στην πρώτη θέση εμφανίζεται ο δείκτης ενός κόμβου που συνδέεται με τον v_i, ενώ η δεύτερη θέση συνδέεται με το επόμενο στοιχείο της λίστας (αν υπάρχει και άλλος κόμβος που συνδέεται με τον v_i) ή περιέχει ένα σύμβολο $\wedge$ (αν δεν υπάρχει άλλος κόμβος που συνδέεται με τον v_i).

Παράδειγμα: Για το γράφημα G στην αρχή της παραγράφου, η λίστα γειτονικότητας είναι:



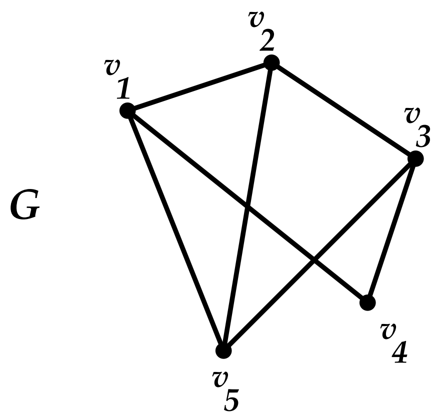
Παρατήρηση: Σε κάθε γραμμή εμφανίζονται όλοι οι κόμβοι που συνδέονται με τον επικεφαλής κόμβο (και όχι κατ' ανάγκη μεταξύ τους).

13. ΑΠΕΙΚΟΝΙΣΗ ΓΡΑΦΗΜΑΤΟΣ ΔΕΣΜΩΝ

Έστω $G = (V, E)$ ένα γράφημα δεσμών. Ορίζουμε την απεικόνιση $\Gamma : V \to \mathcal{P}(V)$ με $\Gamma(v) = \{u \in V : \{v, u\} \in E\}$.

Παρατήρηση: Το ζεύγος (V, Γ) ορίζει το γράφημα G ισοδύναμα με το (V, E) και γι' αυτό μπορούμε να αναφερόμαστε και στο γράφημα (V, Γ) αντί (V, E). Η Γ ονομάζεται **απεικόνιση του γραφήματος δεσμών**.

Παράδειγμα: Στο γράφημα G



αντιστοιχεί η απεικόνιση Γ, με

$$\begin{aligned}
\Gamma(v_1) &= \{v_2, v_4, v_5\},\\
\Gamma(v_2) &= \{v_1, v_3, v_5\},\\
\Gamma(v_3) &= \{v_2, v_4, v_5\},\\
\Gamma(v_4) &= \{v_1, v_3\},\\
\Gamma(v_5) &= \{v_1, v_2, v_3\}.
\end{aligned}$$

Επιπλέον, ορίζουμε τα εξής:

Αν $A = \{v_1, v_2, \ldots, v_k\}$, τότε

$$\Gamma(A) = \Gamma(v_1) \cup \Gamma(v_2) \cup \cdots \cup \Gamma(v_k)$$

και (αναδρομικά) για $n \in \mathbb{N}^*$:

$$\Gamma^n(v) = \Gamma(\Gamma^{n-1}(v)).$$

Παράδειγμα: Για το γράφημα G έχουμε:

Αν $A = \{v_1, v_2\}$ τότε $\Gamma(A) = \{v_1, v_2, v_3, v_4, v_5\}$.

Επίσης, $\Gamma^2(v_1) = \Gamma(\Gamma(v_1)) = \Gamma(\{v_2, v_4, v_5\}) = \{v_1, v_2, v_3, v_5\}$.

14. ΓΡΑΦΗΜΑΤΑ ΜΕ ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ

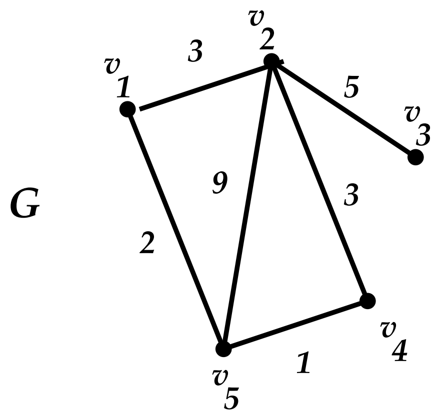
Με ένα γράφημα $G(V, E)$ μπορεί να συσχετισθεί κάποια **συνάρτηση κόστους**, δηλαδή μια συνάρτηση $f : E \to \mathbb{R}$. Οι τιμές $f(\{v_i, v_j\})$ για κάθε $\{v_i, v_j\} \in E$, δίδονται αντίστοιχα και πάνω στο γράφημα.

Βασικές έννοιες, όπως ο ισομορφισμός, τα υπογραφήματα κ.λπ. μεταφέρονται κατά προφανή τρόπο στα γραφήματα με συνάρτηση κόστους.

Προφανώς, μπορούμε επίσης να ορίσουμε και την αντίστοιχη μήτρα γειτονικότητας:

$$M = [m_{ij}], \text{ με } m_{ij} = \begin{cases} f(\{v_i, v_j\}), & \text{αν } \{v_i, v_j\} \in E \\ 0, & \text{αν } \{v_i, v_j\} \notin E. \end{cases}$$

Παράδειγμα: Στο γράφημα G



αντιστοιχεί η μήτρα

$$M = \begin{bmatrix} 0 & 3 & 0 & 0 & 2 \\ 3 & 0 & 5 & 3 & 9 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 1 \\ 2 & 9 & 0 & 1 & 0 \end{bmatrix}.$$

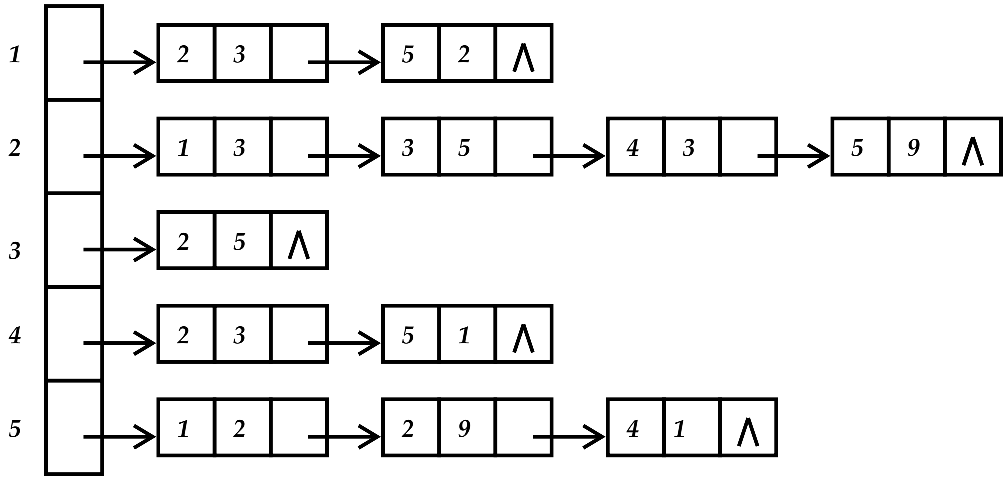
Παρατήρηση: Προφανώς τα γραφήματα χωρίς συνάρτηση κόστους μπορούν να θεωρηθούν σαν ειδική περίπτωση, όπου $f(\{v_i, v_j\}) = 1$, για κάθε $\{v_i, v_j\} \in E$.

Στην περίπτωση που το γράφημα έχει συνάρτηση κόστους, τα στοιχεία της λίστας έχουν την μορφή

κόμβος	κόστος	δείκτης

όπου στη δεύτερη θέση εμφανίζεται το κόστος του αντίστοιχου δεσμού.

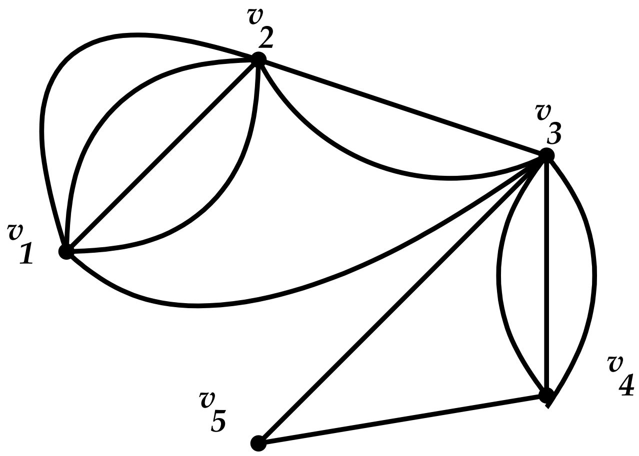
Παράδειγμα: Για το γράφημα G η λίστα γειτονικότητας είναι:



p-γραφήματα δεσμών

Το διατεταγμένο ζεύγος (X, F), όπου $X \neq \emptyset$ και F είναι μια οικογένεια ζευγών $\{v, u\}$ με $v, u \in X$ τέτοια ώστε κανένα στοιχείο της δεν εμφανίζεται περισσότερες από p φορές, λέγεται p-**γράφημα δεσμών**.

Παράδειγμα : Το παρακάτω γράφημα είναι ένα 4-γράφημα δεσμών (αφού το $\{v_1, v_2\}$ εμφανίζεται 4 φορές).



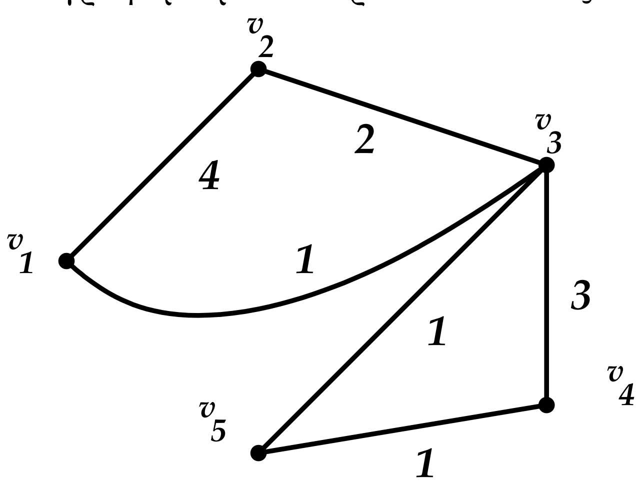
Η μήτρα του παραπάνω 4-γραφήματος είναι προφανώς η

$$M = \begin{bmatrix} 0 & 4 & 1 & 0 & 0 \\ 4 & 0 & 2 & 0 & 0 \\ 1 & 2 & 0 & 3 & 1 \\ 0 & 0 & 3 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

Παρατηρήσεις :

1. Για $p = 1$ παίρνουμε τα **απλά γραφήματα δεσμών** με τα οποία και συνήθως ασχολούμαστε.

2. Προφανώς τα p-γραφήματα δεσμών μπορούν να ταυτιστούν με τα γραφήματα με συνάρτηση κόστους, αρκεί κάθε πολλαπλή εμφάνιση δεσμών μεταξύ δύο κόμβων να αντικατασταθεί με έναν απλό δεσμό ανάμεσα στις κορυφές αυτές, με κόστος ίσο με το πλήθος των πολλαπλών δεσμών. Το παραπάνω γράφημα λοιπόν μπορεί να αντικατασταθεί με το γράφημα με συνάρτηση κόστους



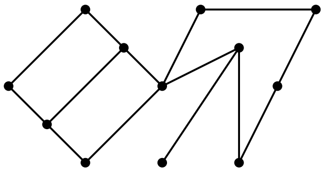
(στο οποίο προφανώς αντιστοιχεί η ίδια μήτρα M).

Ασκήσεις προς επίλυση

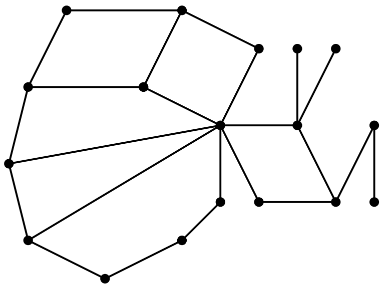
(1) **(*)** Έστω ένα d-κανονικό διμερές γράφημα G και X_1, X_2 η διαμέριση των κορυφών του. Να δειχθεί ότι $|X_1| = |X_2|$.

(2) Να εξετασθεί ποιο από τα παρακάτω γραφήματα είναι διμερές.

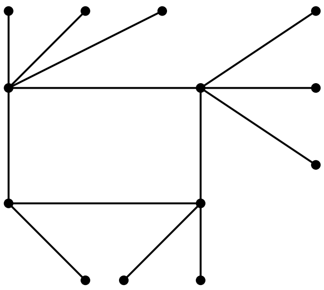
G_1



G_2

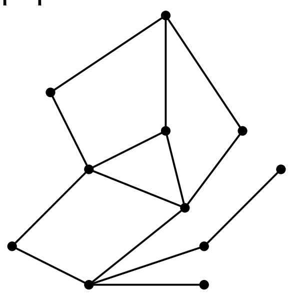


G_3

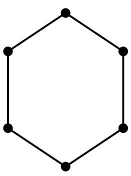


(3) Να επιβεβαιωθεί ο τύπος του Euler για τα παρακάτω γραφήματα:

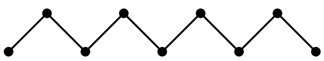
G_1



G_2



G_3



(4) i) Να δειχθεί ότι κάθε γράφημα με 8 κορυφές και περισσότερους από 18 δεσμούς δεν είναι επίπεδο.

ii) Να δειχθεί ότι κάθε διμερές γράφημα με 12 κορυφές και περισσότερους από 20 δεσμούς δεν είναι επίπεδο.

(5) Να δειχθεί ότι σε ένα επίπεδο γράφημα $G = (V, E)$ υπάρχει κορυφή με βαθμό μικρότερο ή ίσο του 5.

Λύση. Έστω ότι ο ελάχιστος βαθμός $\delta(G)$ είναι μεγαλύτερος ή ίσος του 6, τότε

$$2|E| = \sum_{v \in V} d(v) \geq \sum_{v \in V} \delta(G) \geq 6|V(G)| \Leftrightarrow 3|V| \leq |E|$$

Όμως, από το Πόρισμα 29 έχουμε ότι

$$|E| \leq 3|V| - 6$$

Επομένως,

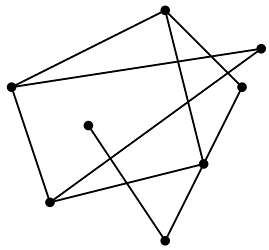
$$3|V| \leq 3|V| - 6$$

άτοπο. Άρα, $\delta(G) \leq 5$. □

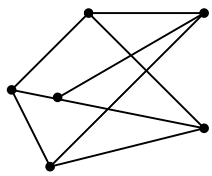
(6) Να κατασκευασθεί ένα (απλό) επίπεδο γράφημα στο οποίο ο ελάχιστος βαθμός των κορυφών του είναι 5.

(7) Να εξετασθεί ποιο από τα παρακάτω γραφήματα είναι επίπεδο.

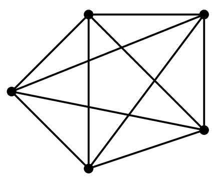
G_1



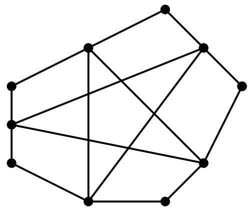
G_2



G_3



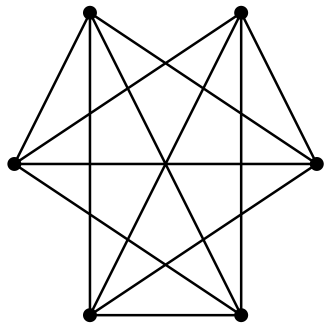
G_4



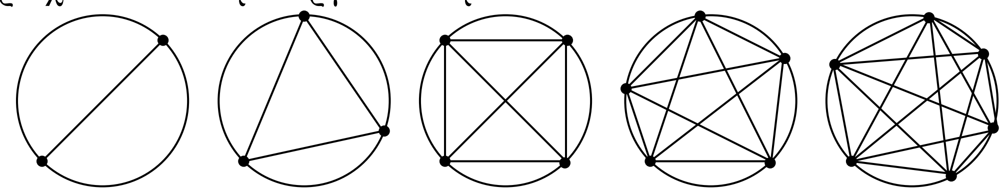
(8) Να εξετασθεί αν κάποιο από τα παρακάτω γραφήματα είναι επίπεδο.

(α) K_6, (β) $K_{5,2}$, (γ) $K_{4,3}$.

(9) Να εξετασθεί αν το παρακάτω γράφημα είναι επίπεδο.



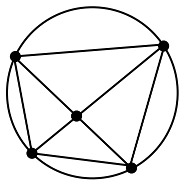
(10) Δίνονται n σημεία πάνω στην περιφέρεια ενός κύκλου σε γενική θέση,[1] Αν ενώσουμε ανά δύο όλα τα σημεία μεταξύ τους, να βρεθεί ο αριθμός a_n των περιοχών που δημιουργούνται μέσα στον κύκλο.



$a_2 = 2$ $\quad a_3 = 4$ $\quad a_4 = 8$ $\quad a_5 = 16$ $\quad a_6 = 31$

Λύση. Προκειμένου να δημιουργήσουμε ένα επίπεδο γράφημα θεωρούμε εκτός από τα n σημεία της περιφέρειας του κύκλου και τα επιπλέον σημεία που προκύπτουν από τις τομές των ευθύγραμμων τμημάτων που τα ενώνουν. Το γράφημα G που προκύπτει είναι ένα επίπεδο γράφημα.

Κάθε 4-άδα σημείων στην περιφέρεια του κύκλου ορίζει ένα μοναδικό σημείο τομής στο εσωτερικό του και αντιστρόφως.



Επομένως, ο αριθμός των επιπλέον σημείων ισούται με $\binom{n}{4}$.

Άρα, $|V(G)| = n + \binom{n}{4}$.

Επιπλέον, ο αριθμός των δεσμών $|E(G)|$ του G ισούται με το μισό του αθροίσματος των βαθμών του. Τα n σημεία που βρίσκονται στην περιφέρεια του κύκλου έχουν το καθένα βαθμό $n + 1$ (αφού το καθένα συνδέεται με ευθύγραμμα τμήματα με τα υπόλοιπα $n - 1$ σημεία, καθώς επίσης συνδέεται πάλι με τα γειτονικά του μέσω των 2 τόξων της περιφέρειας του κύκλου). Τα επιπλέον $\binom{n}{4}$ σημεία έχουν βαθμό 4 το καθένα.

Άρα, το συνολικό άθροισμα των βαθμών του G ισούται με

$$2|E(G)| = n(n+1) + 4\binom{n}{4} \Leftrightarrow |E(G)| = \frac{1}{2}n(n+1) + 2\binom{n}{4}$$

Τέλος, ο ζητούμενος αριθμός περιοχών a_n ισούται με $|F(G)| - 1$ (αφαιρούμε την εξωτερική έδρα του γραφήματος G).

[1] Υπενθυμίζεται ότι ένα σύνολο σημείων βρίσκεται σε γενική θέση αν τα ευθύγραμμα τμήματα που τα ενώνουν δεν τέμνονται ανά 3 στο ίδιο σημείο.

Άρα, από τον τύπο του Euler έχουμε ότι

$$|F(G)| = |E(G)| - |V(G)| + 2 \Leftrightarrow$$

$$a_n + 1 = \frac{1}{2}n(n+1) + 2\binom{n}{4} - n - \binom{n}{4} + 2 \Leftrightarrow$$

$$a_n = \frac{1}{2}n(n-1) + \binom{n}{4} + 1$$

Επομένως, ο ζητούμενος αριθμός a_n ισούται με

$$a_n = \binom{n}{4} + \binom{n}{2} + 1.$$

Πράγματι,

$$a_2 = \binom{2}{4} + \binom{2}{2} + 1 = 0 + 1 + 1 = 2$$

$$a_3 = \binom{3}{4} + \binom{3}{2} + 1 = 0 + 3 + 1 = 4$$

$$a_4 = \binom{4}{4} + \binom{4}{2} + 1 = 1 + 6 + 1 = 8$$

$$a_5 = \binom{5}{4} + \binom{5}{2} + 1 = 5 + 10 + 1 = 16$$

$$a_6 = \binom{6}{4} + \binom{6}{2} + 1 = 15 + 15 + 1 = 31$$

$$a_7 = \binom{7}{4} + \binom{7}{2} + 1 = 35 + 21 + 1 = 57, \text{ κ.ο.κ.}$$ □

(11) Να βρεθεί ο αριθμός διασταυρώσεων του K_6. Να επαληθευθεί το αποτέλεσμα.

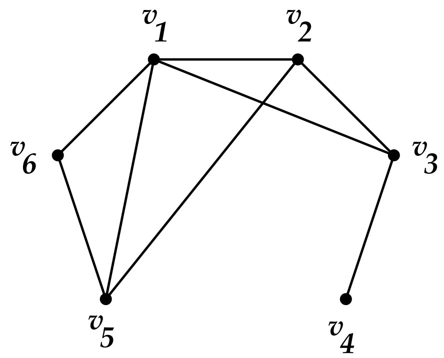
(12) **(*)** Να δειχθεί ότι $cr(K_{3,4}) = 2$.

(13) Να γραφεί το γράφημα με μήτρα:

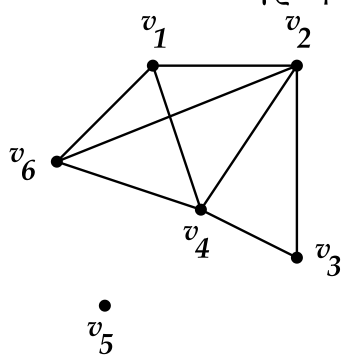
$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

ςδ δ

(14) Να γραφεί η μήτρα και οι λίστες γειτονικότητας του γραφήματος



(15) Να ορισθεί η μήτρα και η απεικόνιση του γραφήματος



(16) Δίδεται ότι η απεικόνιση του γραφήματος G είναι η ακόλουθη: $\Gamma : X \to X$, με

$$\begin{aligned}
\Gamma(v_1) &= \{v_2, v_5\}\\
\Gamma(v_2) &= \{v_1, v_5, v_6\}\\
\Gamma(v_3) &= \{v_4, v_5\}\\
\Gamma(v_4) &= \{v_3, v_5, v_6, v_7\}\\
\Gamma(v_5) &= \{v_1, v_2, v_3, v_4\}\\
\Gamma(v_6) &= \{v_2, v_4, v_7\}\\
\Gamma(v_7) &= \{v_4, v_6\}
\end{aligned}$$

i) Να βρεθούν τα $\Gamma^2(v_1)$, $\Gamma^3(v_7)$.
ii) Να γράφει η μήτρα του G.
iii) Να σχεδιασθεί το G.

(17) Να γραφεί το γράφημα με συνάρτηση κόστους και το το 4-γράφημα δεσμών που έχουν μήτρα:

$$\begin{bmatrix}
0 & 3 & 1 & 0 & 0 & 0 & 2\\
3 & 0 & 2 & 0 & 0 & 0 & 0\\
1 & 2 & 0 & 2 & 0 & 0 & 1\\
0 & 0 & 2 & 0 & 1 & 1 & 4\\
0 & 0 & 0 & 1 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 0 & 0 & 0\\
2 & 0 & 1 & 4 & 0 & 0 & 0
\end{bmatrix}$$

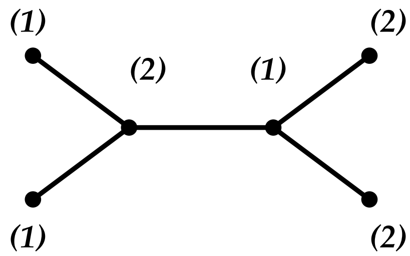
15. ΧΡΩΜΑΤΙΚΟΣ ΑΡΙΘΜΟΣ

Ένα γράφημα ονομάζεται r-**χρωματικό** (r-chromatic) αν οι κορυφές του μπορούν να "χρωματισθούν" με r διαφορετικά χρώματα, έτσι ώστε κάθε δύο ενωμένες κορυφές να έχουν διαφορετικό χρώμα. Κάθε ανάθεση χρωμάτων στις κορυφές με την παραπάνω ιδιότητα ονομάζεται **(έγκυρος) χρωματισμός** ((proper) coloring) του G.

Ο μικρότερος φυσικός r για τον οποίο το γράφημα είναι r-χρωματικό ονομάζεται **χρωματικός αριθμός** (chromatic number) του G, και συμβολίζεται συνήθως με $\chi(G)$.

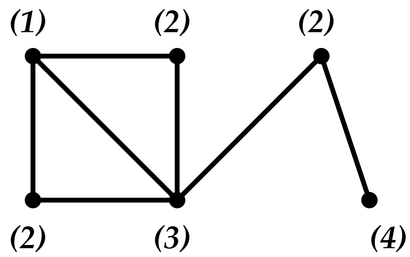
Παράδειγματα :

G_1



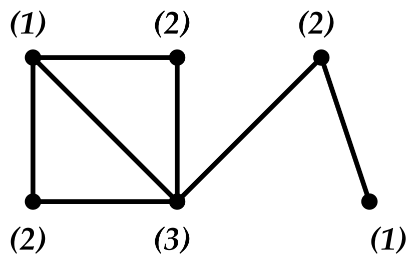
Το γράφημα G_1 είναι 2-χρωματικό με $\chi(G_1) = 2$.

G_2



Το γράφημα G_2 είναι 4-χρωματικό, αλλά $\chi(G_2) = 3$ αφού είναι και 3-χρωματικό (αλλά όχι 2-χρωματικό).

G_2



Παρατηρήσεις

(1) Προφανώς, για κάθε γράφημα G με n κορυφές ισχύει ότι $1 \leq \chi(G) \leq n$. Ειδικότερα, για κάθε $n \geq 1$, ισχύει ότι $\chi(K_n) = n$ και $\chi(K_n^c) = 1$.
Επίσης, $\chi(C_n) = \begin{cases} 2, & n \text{ άρτιος}, \\ 3, & n \text{ περιττός}. \end{cases}$

(2) Αν H είναι υπογράφημα του G τότε $\chi(H) \leq \chi(G)$.
Η παρατήρηση αυτή είναι προφανής αλλά χρησιμοποιείται για να βρίσκουμε κάτω φράγματα για τον χρωματικό αριθμό. Για παράδειγμα, αν το G περιέχει

ένα πλήρες υπογράφημα τάξης k, τότε $\chi(G) \geq k$. Επίσης, αν το G περιέχει ένα κύκλο περιττού μήκους, τότε $\chi(G) \geq 3$.

Πρόταση 32. *Ένα γράφημα G έχει χρωματικό αριθμό 2 ανν είναι διμερές.*

Παρατήρηση. Με βάση τις παραπάνω παρατηρήσεις γνωρίζουμε πότε ένα γράφημα έχει χρωματικό αριθμό 1, 2 και n. Δεν γνωρίζουμε όμως κάποια αντίστοιχη πρόταση που να χαρακτηρίζει τα γραφήματα με χρωματικό αριθμό 3. Επίσης, στην γενική περίπτωση, ο υπολογισμός του χρωματικού αριθμού ενός γραφήματος είναι υπολογιστικά δύσκολος. Το πρόβλημα του χρωματισμού στην γενική περίπτωση είναι NP-complete.

Στην επόμενη πρόταση δίνεται ένα άνω φράγμα για τον χρωματικό αριθμό ενός γραφήματος G με βάση τον μέγιστο βαθμό $\Delta(G)$ των κορυφών του.

Πρόταση 33. *Για κάθε γράφημα G ισχύει ότι $\chi(G) \leq \Delta(G) + 1$.*

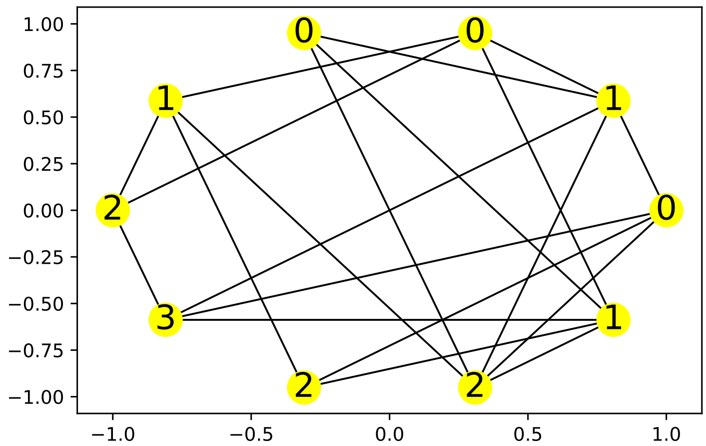
Απόδειξη. Έστω $|V(G)| = n$. Αριθμούμε τις κορυφές του G με οποιαδήποτε σειρά. Χρωματίζουμε την κορυφή v_1 με το χρώμα 1. Στην συνέχεια, αν η v_2 είναι γειτονική με την v_1 την χρωματίζουμε με το χρώμα 2, αλλιώς την χρωματίζουμε με το χρώμα 1. Γενικότερα, έστω ότι έχουμε ήδη χρωματίσει τις κορυφές $v_1, v_2, \ldots, v_k$, όπου $1 \leq k < n$, με χρώματα από το σύνολο $\{1, 2, \ldots, \Delta(G) + 1\}$. Χρωματίζουμε την κορυφή v_{k+1} με το ελάχιστο χρώμα που δεν έχει χρησιμοποιηθεί για τους γείτονές της. Επειδή, $d(v_{k+1}) \leq \Delta(G)$ τουλάχιστον ένα χρώμα είναι διαθέσιμο για την v_{k+1}. Άρα, οι κορυφές του G μπορούν να χρωματισθούν χρησιμοποιώντας το πολύ $\Delta(G)+1$ χρώματα. □

Η απόδειξη της Πρότασης 33 δίνει έναν άπληστο αλγόριθμο για τον χρωματισμό ενός γραφήματος G χρησιμοποιώντας το πολύ $\Delta(G) + 1$ χρώματα. Η βιβλιοθήκη `networkx` διαθέτει την μέθοδο `greedy_color(G,strategy='largest_first')` η οποία υλοποιεί την παραπάνω ιδέα ταξινομώντας τις κορυφές με βάση κάποια στρατηγική, π.χ. σε φθίνουσα σειρά με βάση τους βαθμούς τους, και επιστρέφει ένα λεξικό που περιέχει έναν χρωματισμό. Μια ανεξάρτητη υλοποίηση δίδεται παρακάτω:

```python
import networkx as nx
import matplotlib.pyplot as plt

G = nx.gnm_random_graph(10,20)

#finds the mininum excluded value of a SET of integers
def mex(X):
    for i in range(0,len(X)):
        if i not in X:
            return i
    return len(X)

#empty dictionary
colors = {}
#assign a large default (to be updated) color to all nodes
for u in G:
    colors[u] = len(G)
```

```
#greedy algorithm for coloring using at most D(G) + 1 colors
for u in G:
    nbrs = list(G[u])
    nbrs_colors = set([colors[v] for v in nbrs])
    colors[u] = mex(nbrs_colors)

print(colors.keys())
print(colors.values())

pos = nx.layout.circular_layout(G)
nx.draw_networkx(G,pos,with_labels=True,labels=colors,font_size=18,node_color='yellow
    ')
plt.show()
```

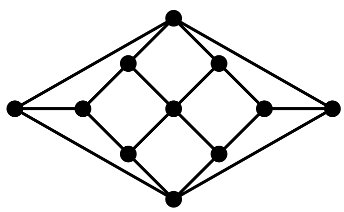
Output:



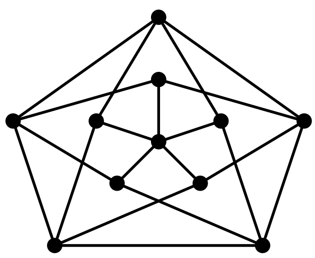
Όπως είδαμε, υπάρχουν γραφήματα για τα οποία $\chi(G) = \Delta(G)+1$. Για παράδειγμα $\chi(K_n) = n = \Delta(K_n) + 1$, και $\chi(C_r) = 3 = \Delta(C_r) + 1$ όπου $r \geq 3$ περιττός. Ο Rowland Leonard Brooks απέδειξε ότι αυτά είναι τα μόνα συνεκτικά γραφήματα που έχουν αυτή την ιδιότητα.

Πρόταση 34 (Brooks, 1941). *Αν G είναι ένα συνεκτικό γράφημα με $G \neq K_n$ και $G \neq C_r$, όπου $r \geq 3$ περιττός, τότε $\chi(G) \leq \Delta(G)$.*

Άσκηση 1. *Να βρεθεί ο χρωματικός αριθμός των παρακάτω γραφημάτων:*



γράφημα Herschel



γράφημα Grötzsch

Το πρόβλημα του χρωματισμού των κορυφών ενός γραφήματος ανακύπτει με φυσικό τρόπο κάθε φορά που προσπαθούμε να αντιστοιχίσουμε πόρους υπό την παρουσία διενέξεων.[2]

(1) (Προγραμματισμός διεργασιών) Υποθέστε ότι έχουμε μια συλλογή n διεργασιών σε ένα σύστημα που μπορεί να εκτελεί πολλές εργασίες ταυτόχρονα, όμως ορισμένα ζευγάρια διεργασιών δεν μπορούν να προγραμματισθούν για την ίδια χρονική περίοδο επειδή χρειάζονται και τα δύο ένα συγκεκριμένο πόρο του συστήματος. Για τα επόμενα k χρονικά βήματα του συστήματος, θα θέλαμε να προγραμματίσουμε κάθε διεργασία ώστε να εκτελεστεί σε τουλάχιστον έναν από αυτούς τους πόρους. Είναι δυνατόν αυτό; Αν κατασκευάσουμε ένα γράφημα G με κορυφές τις διεργασίες, συνδέοντάς τις με δεσμό όταν παρουσιάζουν διένεξη, τότε ένας χρωματισμός με k χρώματα του G αντιπροσωπεύει ένα χρονοδιάγραμμα των διεργασιών χωρίς διενέξεις. Όλες οι κορυφές που έχουν το χρώμα j μπορούν να προγραμματισθούν στο βήμα j, και δεν θα υπάρξει ποτέ ανταγωνισμός για κανένα πόρο.
(2) (Μεταγλώττιση κώδικα) Υποθέστε ότι μεταγλωττίζουμε ένα πρόγραμμα και προσπαθούμε να αποδώσουμε κάθε μεταβλητή σε ένα από k καταχωρητές (registers). Αν δύο μεταβλητές είναι ταυτόχρονα σε χρήση μια δεδομένη χρονική στιγμή, τότε δεν μπορούν να αντιστοιχιστούν στον ίδιο καταχωρητή. (Διαφορετικά, η μια θα καταλήξει να αντικαταστήσει την άλλη.) Μπορούμε να κατασκευάσουμε ένα γράφημα G με κορυφές τις μεταβλητές, συνδέοντας τις μεταβλητές με ένα δεσμό αν χρησιμοποιούνται και οι δύο την ίδια χρονική στιγμή. Ένας χρωματισμός του G με k χρώματα αντιστοιχεί σε έναν ασφαλή τρόπο αντιστοίχισης μεταβλητών στους καταχωρητές. Όλες οι κορυφές που έχουν το χρώμα j μπορούν να αντιστοιχιστούν στον καταχωρητή j, αφού ανά δύο δεν χρησιμοποιούνται την ίδια χρονική στιγμή.
(3) (Απόδοση συχνοτήτων σε συσκευές ασύρματης επικοινωνίας) Υποθέστε ότι θέλουμε να αποδώσουμε ένα από τα k διαθέσιμα μήκη κύματος μετάδοσης σε κάθε μια από n συσκευές. Αν όμως δύο συσκευές είναι αρκετά κοντά η μία στην άλλη, τότε θα πρέπει να τους αποδοθούν διαφορετικά μήκη κύματος (συχνοτήτων) ώστε να αποφευχθούν οι παρεμβολές. Για να το αντιμετωπίσουμε κατασκευάζουμε ένα γράφημα G με κορυφές τις συσκευές, συνδέοντας δύο κορυφές αν είναι αρκετά κοντά ώστε να παρεμβάλλεται η μια στην άλλη. Ένας χρωματισμός του G με k χρώματα είναι μια απόδοση συχνοτήτων έτσι ώστε κάθε κόμβος στον οποίο αποδίδεται το ίδιο μήκος κύματος να είναι αρκετά μακριά ώστε να μην δημιουργεί πρόβλημα παρεμβολών.

[2] Τα παραδείγματα προέρχονται από την ελληνική έκδοση του βιβλίου [;].

Άσκηση 2. *Οκτώ χημικές ουσίες $c_1, c_2, \ldots, c_8$ πρόκειται να μεταφερθούν με αεροπλάνο από μια τοποθεσία σε μια άλλη. Το κόστος μεταφοράς εξαρτάται από τον αριθμό των συσκευασιών που θα χρησιμοποιηθούν για την αποστολή. Το κόστος αποστολής μια συσκευασίας είναι 100 ευρώ. Για κάθε επιπλέον συσκευασία το κόστος προσαυξάνεται κατά 70 ευρώ. Κάποιες χημικές ουσίες αλληλεπιδρούν μεταξύ τους και είναι επικίνδυνο να μεταφερθούν στην ίδια συσκευασία. Οι χημικές ουσίες που αλληλεπιδρούν είναι οι εξής:*

$$\begin{array}{lll} c_1 : c_2, c_5, c_6 & c_2 : c_1, c_3, c_5, c_7 & c_3 : c_2, c_4, c_7 \\ c_4 : c_3, c_6, c_7, c_8 & c_5 : c_1, c_2, c_6, c_7, c_8 & c_6 : c_1, c_4, c_5, c_8 \\ c_7 : c_2, c_3, c_4, c_5, c_8 & c_8 : c_4, c_5, c_6, c_7. & \end{array}$$

Ποιό είναι το ελάχιστο κόστος αποστολής των οκτώ χημικών ουσιών και με ποιά τοποθέτηση στις συσκευασίες επιτυγχάνεται αυτό;

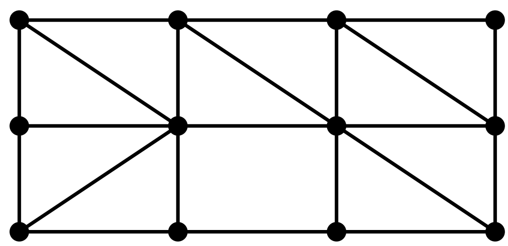
Χρωματικός αριθμός επίπεδων γραφημάτων.

Μετά από περισσότερο από 100 χρόνια προσπαθειών αποδείχθηκε η επόμενη πρόταση σχετικά με τον χρωματικό αριθμό των επίπεδων γραφημάτων.

Πρόταση 35 (Appel, Haken, Koch, 1976). *Αν G είναι επίπεδο γράφημα, τότε $\chi(G) \leq 4$.*

Η απόδειξη των Appel, Haken και Koch δημιούργησε διαμάχες στην εποχή της: Ενώ η δομή της απόδειξης είναι μια απλή επαγωγή ως προς τον αριθμό των εδρών του γραφήματος, το βήμα της επαγωγής περιλαμβάνει σχεδόν δύο χιλιάδες πολύπλοκες περιπτώσεις και η επαλήθευση αυτών των περιπτώσεων έπρεπε να πραγματοποιηθεί από υπολογιστή. Αυτό δεν ήταν ικανοποιητικό αποτέλεσμα για τους περισσότερους μαθηματικούς διότι ενώ έλπιζαν για μια απόδειξη που θα τους εξηγούσε γιατί ισχύει το αποτέλεσμα, αντί γι' αυτό πήραν μια ανάλυση περιπτώσεων με τεράστια πολυπλοκότητα. Το πρόβλημα της εύρεσης μια σχετικά σύντομης και κατανοητής απόδειξης είναι ακόμα ανοιχτό.

Άσκηση 3. *Να βρεθεί ο χρωματικός αριθμός του γραφήματος*



(Υπόδειξη: Να δειχθεί ότι 3 χρώματα δεν επαρκούν για να χρωματίσουν το γράφημα.)

Χρωματικό πολυώνυμο.

Έστω G ένα γράφημα δεσμών. Ο αριθμός των διαφορετικών χρωματισμών του G με λ χρώματα συμβολίζεται με $\chi_G(\lambda)$ ή $\chi(G;\lambda)$.

Για παράδειγμα, $\chi(K_n;\lambda) = \lambda(\lambda-1)(\lambda-2)\cdots(\lambda-n+1)$. Επίσης, $\chi(K_n^c;\lambda) = \lambda^n$. Προφανώς, για κάθε $\lambda < \chi(G)$ ισχύει ότι $\chi(G;\lambda) = 0$.

Αποδεικνύεται ότι για κάθε γράφημα G με n κορυφές και m δεσμούς ο αριθμός $\chi(G;\lambda)$ είναι πολυώνυμο του λ με βαθμό n. Ο συντελεστής του λ^n ισούται με 1, ο συντελεστής του λ^{n-1} ισούται με $-m$, ο συντελεστής του λ^{n-2} είναι ίσος με $m(m-1)/2$ μείον τον αριθμό των τριγώνων του G και ότι το πολυώνυμο δεν έχει σταθερό όρο.

Άμεσα προκύπτει ότι αν ένα γράφημα G αποτελείται από k συνεκτικές συνιστώσες $G_1, G_2, \ldots, G_k$ τότε $\chi(G;\lambda) = \chi(G_1;\lambda)\chi(G_2;\lambda)\cdots\chi(G_k;\lambda)$.

Πρόταση 36. *Να δειχθεί ότι για κάθε δένδρο*[3] *T με n κορυφές ισχύει ότι*

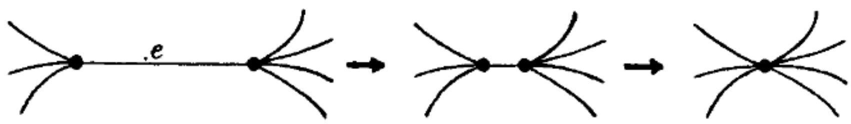
$$\chi(T;\lambda) = \lambda(\lambda-1)^{n-1}.$$

Λύση. Θα χρησιμοποιήσουμε επαγωγή ως προς τον αριθμό n των κορυφών του T. Για $n = 1$ ο τύπος ισχύει, αφού $\chi(T;\lambda) = \lambda$. Έστω ότι ισχύει για κάθε δένδρο T με $n-1$ κορυφές δηλαδή $\chi(T;\lambda) = \lambda(\lambda-1)^{n-2}$.

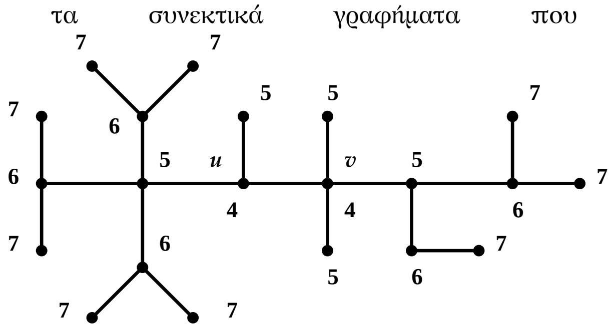
Κάθε δένδρο T με n κορυφές περιέχει τουλάχιστον μια κορυφή βαθμού 1 (φύλλο). Σβήνοντας την κορυφή αυτή προκύπτει ένα δένδρο T' με $n-1$ κορυφές το οποίο μπορεί να χρωματισθεί με $\lambda(\lambda-1)^{n-2}$ τρόπους. Αν προσθέσουμε ξανά την κορυφή που σβήσαμε τότε αυτή έχει $\lambda-1$ τρόπους να χρωματισθεί, οπότε συνολικά έχουμε $\lambda(\lambda-1)^{n-1}$ τρόπους χρωματισμού του T. □

Παρατηρήστε ότι όλα τα δένδρα με n κορυφές, ανεξάρτητα από τον τρόπο που συνδέονται αυτές, έχουν τον ίδιο αριθμό χρωματισμών με λ χρώματα.

Έστω $G = (V, E)$ και $e \in E$. Ορίζουμε δύο γραφήματα G'_e και G''_e με σύνολο δεσμών $E(G) \setminus \{e\}$. Το πρώτο γράφημα G'_e προκύπτει από το G αφαιρώντας τον δεσμό e. Το δεύτερο γράφημα G''_e προκύπτει από το G ταυτίζοντας τα άκρα του δεσμού e και αφαιρώντας τον e. Η πράξη αυτή ονομάζεται **σύνθλιψη** (contraction) του δεσμού e.



[3]Δένδρα ονομάζονται τα συνεκτικά γραφήματα που δεν έχουν κύκλους.



Θα μελετήσουμε τα δένδρα αναλυτικά σε επόμενες διαλέξεις.

Μπορούμε να υπολογίσουμε το χρωματικό πολυώνυμο ενός γραφήματος αναδρομικά χρησιμοποιώντας την επόμενη πρόταση.

Πρόταση 37. *Έστω G ένα γράφημα δεσμών και $e \in E(G)$. Τότε*

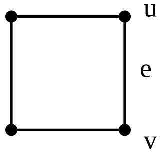
$$\chi(G;\lambda) = \chi(G'_e;\lambda) - \chi(G''_e;\lambda).$$

Απόδειξη. Αν e είναι ένας δεσμός του G μπορούμε να διαμερίσουμε τους χρωματισμούς του G'_e σε δύο κατηγορίες: Αυτούς που τα δύο άκρα του έχουν διαφορετικά χρώματα (οπότε είναι και έγκυροι χρωματισμοί του G) και σ' αυτούς που τα δύο άκρα του έχουν το ίδιο χρώμα (οπότε είναι έγκυροι χρωματισμοί του G''_e). Άρα, $\chi(G'_e;\lambda) = \chi(G;\lambda) + \chi(G''_e;\lambda)$ □

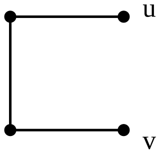
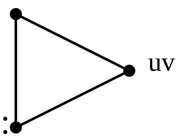
Άσκηση 4. *Να υπολογισθεί το χρωματικό πολυώνυμο του γραφήματος C_4.*

Απάντηση. $\chi(C_4;\lambda) = \lambda^4 - 4\lambda^3 + 6\lambda^2 - 3\lambda$.

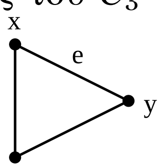
Λύση. Έστω e ένας οποιοσδήποτε δεσμός του C_4



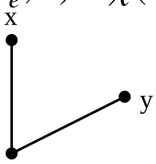
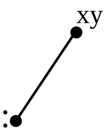
τότε έχουμε ότι $\chi(C_4;\lambda) = \chi(G'_e;\lambda) - \chi(G''_e;\lambda)$ όπου

G'_e:  και $G''_e = C_3$: 

Επειδή το G'_e είναι δένδρο με 4 κορυφές έπεται ότι $\chi(G'_e;\lambda) = \lambda(\lambda-1)^3$.
Έστω e ένας οποιοσδήποτε δεσμός του C_3



τότε έχουμε ότι $\chi(C_4;\lambda) = \chi(G'_e;\lambda) - \chi(G''_e;\lambda)$ όπου

G'_e:  και G''_e: 

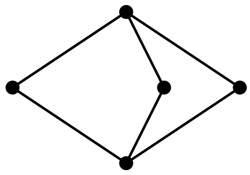
Επειδή τα G'_e και G''_e είναι δένδρα με 3 και 2 κορυφές αντίστοιχα έπεται ότι $\chi(G'_e;\lambda) = \lambda(\lambda-1)^2$ και $\chi(G''_e;\lambda) = \lambda(\lambda-1)$.

Άρα, τελικά, έχουμε ότι

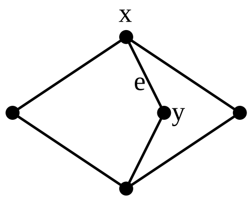
$$\begin{aligned}\chi(C_4;\lambda) &= \lambda(\lambda-1)^3 - (\lambda(\lambda-1)^2 - \lambda(\lambda-1)) = \lambda(\lambda-1)\left((\lambda-1)^2 - (\lambda-1-1)\right)\\ &= \lambda(\lambda-1)(\lambda^2-3\lambda+3)\\ &= \lambda^4 - 4\lambda^3 + 6\lambda^2 - 3\lambda\end{aligned}$$

Επομένως, υπάρχουν $\chi(C_4;10) = 10^4 - 4\cdot10^3 + 6\cdot10^2 - 3\cdot10 = 6570$ τρόποι χρωματισμού του C_4 με 10 χρώματα. □

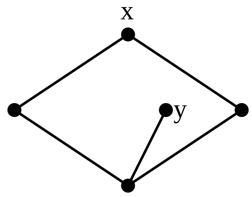
Άσκηση 5. *Να υπολογισθεί το χρωματικό πολυώνυμο του γραφήματος G*

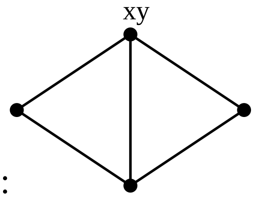


Λύση. Έστω e ο παρακάτω δεσμός του γραφήματος G



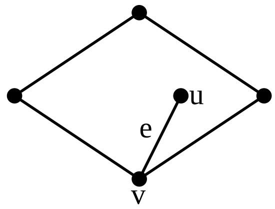
τότε έχουμε ότι $\chi(G;\lambda) = \chi(G'_e;\lambda) - \chi(G''_e;\lambda)$ όπου

$G_1 = G'_e$: 

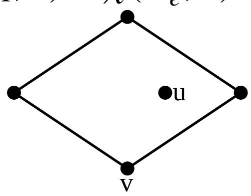
και $G_2 = G''_e$: 

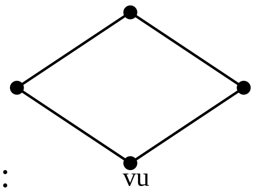
Για καθένα από τα παραπάνω δύο γραφήματα G_1, G_2 έχουμε αντίστοιχα:

- Έστω e ο παρακάτω δεσμός του γραφήματος G_1



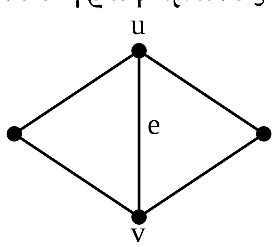
τότε έχουμε ότι $\chi(G_1;\lambda) = \chi(G'_e;\lambda) - \chi(G''_e;\lambda)$ όπου

$G'_e = C_4 \cup K_1$: 

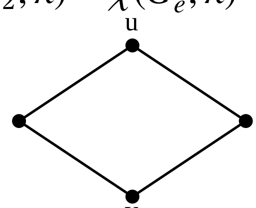
και $G''_e = C_4$: 

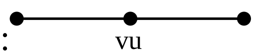
οπότε $\chi(G_1;\lambda) = \chi(K_1;\lambda)\chi(C_4;\lambda) - \chi(C_4;\lambda) = (\lambda-1)\chi(C_4;\lambda)$

- Έστω e ο παρακάτω δεσμός του γραφήματος G_2



τότε έχουμε ότι $\chi(G_2;\lambda) = \chi(G'_e;\lambda) - \chi(G''_e;\lambda)$ όπου

$G'_e = C_4$: 

και G''_e: 

οπότε $\chi(G_2;\lambda) = \chi(C_4;\lambda) - \lambda(\lambda-1)^2$.

Άρα, τελικά, έχουμε ότι

$$\begin{aligned}\chi(G;\lambda) &= (\lambda-1)\chi(C_4;\lambda) - \chi(C_4;\lambda) + \lambda(\lambda-1)^2 \\ &= (\lambda-2)\lambda(\lambda-1)(\lambda^2-3\lambda+3) + \lambda(\lambda-1)^2\end{aligned}$$

Για παράδειγμα, υπάρχουν $\chi(G;10) = 8\cdot 10\cdot 9\cdot(10^2 - 10\cdot 10 + 3) + 10\cdot 9^2 = 53370$ τρόποι χρωματισμού του G με 10 χρώματα. □

Άσκηση 6 (*). *Να υπολογισθεί το χρωματικό πολυώνυμο του κύκλου* C_n, $n \geq 3$.

Απάντηση. $\chi(C_n;\lambda) = (\lambda-1)^n + (-1)^n(\lambda-1)$, $n \geq 3$.

Άσκηση 7. *Να δειχθεί ότι για κάθε γράφημα* G *ισχύει ότι*

$$\chi(K_1 + G;\lambda) = \lambda\cdot\chi(G;\lambda-1).$$

Άσκηση 8. *Να υπολογισθεί το χρωματικό πολυώνυμο του "τροχού"* W_n, $n \geq 3$, *(ο οποίος προκύπτει από το* C_n *προσθέτοντας μια νέα κορυφή η οποία ενώνεται με όλες τις κορυφές του κύκλου* C_n*).*
(Υπόδειξη: Χρησιμοποιήστε τις προηγούμενες δύο ασκήσεις)

Απάντηση. $\chi(W_n;\lambda) = \lambda(\lambda-2)^n + (-1)^n\lambda(\lambda-2)$, $n \geq 3$.

Άσκηση 9. *Να δειχθεί ότι για κάθε ζεύγος γραφημάτων* G, H *ισχύει ότι*

$$\chi(G \cup H;\lambda) = \chi(G;\lambda)\chi(H;\lambda).$$

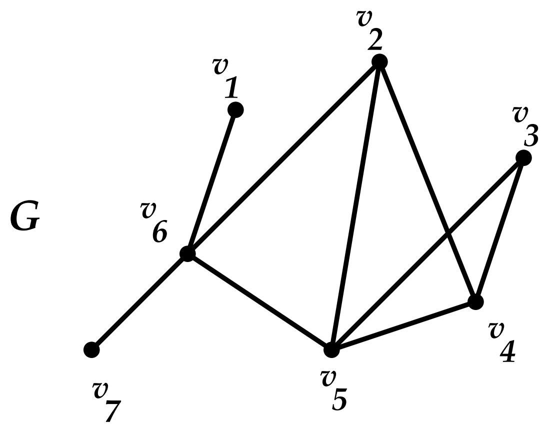
16. ΑΝΕΞΑΡΤΗΣΙΑ - ΚΑΛΥΨΗ

Ένα σύνολο κορυφών $A \subseteq V$ σε ένα γράφημα $G = (V, E)$ ονομάζεται **ανεξάρτητο** (independent) ανν

$$\text{για κάθε } v, u \in A \text{ ισχύει ότι } \{v, u\} \notin E$$

(δηλαδή αν οποιεσδήποτε δύο κορυφές του δεν συνδέονται μεταξύ τους). Τα στοιχεία του A λέγονται **ανεξάρτητες** κορυφές. Ο μέγιστος αριθμός ανεξάρτητων κορυφών του G λέγεται **δείκτης ανεξαρτησίας κορυφών** του G.

Παράδειγμα: Έστω το γράφημα



Το σύνολο $\{v_1, v_2, v_3, v_4\}$ δεν είναι ανεξάρτητο, διότι οι κορυφές v_3, v_4 συνδέονται στο G. Το σύνολο $\{v_1, v_5, v_7\}$ είναι ανεξάρτητο, αλλά ο δείκτης ανεξαρτησίας κορυφών του G είναι 4, αφού και το σύνολο $\{v_1, v_2, v_3, v_7\}$ είναι ανεξάρτητο, (ενώ δεν υπάρχει ανεξάρτητο υποσύνολο του V με περισσότερα στοιχεία).

Πρόταση 38. *Έστω $G = (V, E)$ ένα γράφημα δεσμών και $A \subseteq V$.*

i). Το A είναι ανεξάρτητο στο G αν και μόνο αν τα στοιχεία του αποτελούν τις κορυφές μιας κλίκας του G^c.

ii). Ο δείκτης ανεξαρτησίας κόμβων του G είναι ίσος με το μέγιστο μέγεθος κλίκας του G^c.

Απόδειξη.

i). Αν το A είναι ανεξάρτητο τότε για κάθε $v, u \in A$ ισχύει ότι $\{v, u\} \notin E(G)$ επομένως $\{v, u\} \in E(G^c)$. Άρα, το σύνολο A αποτελεί κλίκα του G^c, και αντιστρόφως.

ii). Αφού κάθε ανεξάρτητο σύνολο κορυφών A του G είναι κλίκα του G^c και αντιστρόφως, προφανώς κάθε μέγιστο ανεξάρτητο σύνολο κορυφών του G αποτελεί μέγιστη κλίκα του G^c. □

Κάλυμμα από κορυφές (vertex cover) ενός γραφήματος δεσμών $G = (V, E)$ ονομάζεται ένα σύνολο κορυφών $B \subseteq V$, ανν

$$\text{για κάθε } e \in E, \text{ υπάρχει } v \in B \text{ τέτοια ώστε } v \in e,$$

δηλαδή κάθε δεσμός του G καλύπτεται από (τουλάχιστον) μια κορυφή του B. Ο ελάχιστος αριθμός στοιχείων σε ένα κάλυμμα από κορυφές ονομάζεται **δείκτης κάλυψης** του G.

Παράδειγμα: Για το προηγούμενο γράφημα G, το σύνολο $B_1 = \{v_1, v_2, v_3, v_4, v_5\}$ δεν είναι κάλυμμα από κορυφές, αφού ο δεσμός $\{v_6, v_7\}$ δεν καλύπτεται από καμία κορυφή του B_1.

Το σύνολο $B_2 = \{v_2, v_3, v_4, v_6\}$ είναι ένα κάλυμμα από κορυφές, αλλά ο δείκτης κάλυψης του G είναι 3, αφού και το σύνολο $\{v_4, v_5, v_6\}$ είναι ένα κάλυμμα από κορυφές του G με 3 στοιχεία, (ενώ δεν υπάρχει κάλυμμα από κορυφές του G με λιγότερα στοιχεία).

Πρόταση 39. *Έστω $G = (V, E)$ ένα γράφημα δεσμών και $A \subseteq V$.*

i) Το A είναι ανεξάρτητο αν και μόνο αν το $V \setminus A$ είναι κάλυμμα από κορυφές του G.

ii) Το άθροισμα του δείκτη ανεξαρτησίας και του δείκτη κάλυψης του G ισούται με $|V|$.

Απόδειξη.

i) Έστω A ένα ανεξάρτητο σύνολο κορυφών του G, τότε κανένας δεσμός του G δεν περιέχει και τα δύο άκρα του στο A, άρα κάθε δεσμός του G έχει τουλάχιστον ένα από τα δύο άκρα του στο σύνολο $V \setminus A$, δηλαδή το σύνολο $V \setminus A$ είναι κάλυμμα από κορυφές του G. Αντίστροφα, αν το σύνολο B είναι κάλυμμα από κορυφές του G, τότε δεν υπάρχει δεσμός του G που να έχει και τα δύο άκρα του στο $V \setminus B$, διότι τότε δεν θα καλύπτονταν από το B, άρα το σύνολο $V \setminus B$ είναι ανεξάρτητο σύνολο κορυφών του G.

ii) Αν A είναι ένα ανεξάρτητο σύνολο κορυφών του G τότε το $V \setminus A$ είναι ένα κάλυμμα κορυφών του G και αντιστρόφως. Άρα, σε κάθε μέγιστο ανεξάρτητο σύνολο κορυφών του G αντιστοιχεί ένα ελάχιστο κάλυμμα κορυφών του G.

□

Παρατήρηση: Η εύρεση ενός μέγιστου ανεξάρτητου συνόλου κορυφών του G ονομάζεται πρόβλημα ανεξάρτητου συνόλου (IndependentSet problem). Η εύρεση μια μέγιστης κλίκας σε ένα γράφημα ονομάζεται πρόβλημα κλίκας (Clique problem). Η εύρεση ενός ελάχιστου καλύμματος κορυφών του G ονομάζεται πρόβλημα καλύμματος από κορυφές (VertexCover problem). Οι προτάσεις 38 και 39 αποτελούν παραδείγματα αναγωγής (reduction) ή μετασχηματισμού (transformation) ενός προβλήματος σε άλλο ισοδύναμο πρόβλημα. Δείχνουν ότι μπορούμε να λύσουμε οποιοδήποτε από τα τρία προβλήματα IndependentSet, Clique και VertexCover χρησιμοποιώντας αλγορίθμους που λύνουν το ένα από τα αυτά. Επίσης, δείχνουν ότι τα τρία προβλήματα έχουν την ίδια υπολογιστική δυσκολία.

Τα προβλήματα IndependentSet, Clique και VertexCover ανήκουν σε μια κλάση προβλημάτων τα οποία θεωρούνται υπολογιστικά δύσκολα και ονομάζονται NP-Complete. Η κλάση αυτή αποτελείται από δύσκολα[4] υπολογιστικά προβλήματα για τα οποία υπάρχουν αναγωγές από το ένα στο άλλο, έτσι ώστε αν κάποιο από αυτά λυθεί με γρήγορο τρόπο να μπορούν να λυθούν με γρήγορο τρόπο όλα τα υπόλοιπα προβλήματα της κλάσης.

[4] Οι γνωστοί αλγόριθμοι επίλυσης απαιτούν εκθετικό αριθμό βημάτων σε σχέση με το μέγεθος εισόδου.

Η ανάγκη επίλυσης αυτών των προβλημάτων δημιούργησε πολλές νέες ιδέες, τεχνικές και μεθοδολογίες λύσης προβλημάτων. Ενδεικτικά αναφέρονται οι έννοιες των προσεγγιστικών αλγορίθμων, ευρετικών τεχνικών, τυχαιοποιημένων αλγορίθμων, παραμετρικής πολυπλοκότητας. Οι ιδέες αυτές φάνηκαν χρήσιμες και σε άλλα απλούστερα προβλήματα και δημιούργησαν μια άνθηση στον κλάδο της σχεδίασης αλγορίθμων.

Στην ενότητα αυτή παρουσιάζονται, μέσω παραδειγμάτων, δύο από τις τεχνικές που δημιουργήθηκαν για την αντιμετώπιση τέτοιων δύσκολων προβλημάτων.

Προσεγγιστικοί αλγόριθμοι. Η πρώτη τεχνική ονομάζεται προσεγγιστικός αλγόριθμος. Οι προσεγγιστικοί αλγόριθμοι δημιουργήθηκαν ως συμβιβαστική λύση στην αδυναμία να λύσουμε ένα δύσκολο πρόβλημα. Το βασικό τους χαρακτηριστικό είναι ότι αν και δεν υπολογίζουν πάντα την βέλτιστη λύση, εν τούτοις εξασφαλίζουν ότι η λύση που βρίσκουμε θα είναι χειρότερη από την βέλτιστη το πολύ κατά κάποιο ποσοστό, το οποίο είναι γνωστό εκ των προτέρων. Για παράδειγμα, αν και δεν μπορούμε να βρούμε το ελάχιστο κάλυμμα κορυφών σε ένα γράφημα μήπως μπορούμε να βρούμε ευκολότερα ένα κάλυμμα κορυφών που να μην είναι πολύ μεγάλο σε σχέση με το ελάχιστο;

Έστω ότι για ένα πρόβλημα η βέλτιστη λύση έχει μέγεθος ή τιμή OPT. Ένας ϵ-**προσεγγιστικός αλγόριθμος** για το πρόβλημα αυτό είναι ένας αλγόριθμος που βρίσκει μια λύση με μέγεθος ή τιμή a όπου

$$\frac{|a - \mathrm{OPT}|}{\mathrm{OPT}} \leq \epsilon,$$

δηλαδή η προσεγγιστική λύση που βρίσκουμε έχει **σχετικό σφάλμα** το πολύ ϵ από την πραγματική λύση. (**Προσοχή!** Υπάρχουν πολλοί συγγραφείς για τους οποίους ο συμβολισμός ϵ-προσεγγιστικός αλγόριθμος σημαίνει ότι $\frac{a}{OPT} \leq \epsilon$ αν πρόκειται για πρόβλημα ελαχιστοποίησης και $\frac{OPT}{a} \leq \epsilon$ αν πρόκειται για πρόβλημα μεγιστοποίησης. Επίσης, υπάρχουν αρκετές ακόμα παραλλαγές στους συμβολισμούς.)

Σε προβλήματα ελαχιστοποίησης ισχύει ότι $a \geq \mathrm{OPT}$, ενώ σε προβλήματα μεγιστοποίησης ισχύει ότι $a \leq \mathrm{OPT}$.

Για παράδειγμα ένας 2-προσεγγιστικός αλγόριθμος για ένα πρόβλημα ελαχιστοποίησης βρίσκει μια λύση με μέγεθος a όπου

$$\frac{a - \mathrm{OPT}}{\mathrm{OPT}} \leq 2 \Leftrightarrow a - \mathrm{OPT} \leq 2 \cdot \mathrm{OPT} \Leftrightarrow a \leq 3 \cdot \mathrm{OPT},$$

δηλαδή βρίσκει μια λύση που έχει μέγεθος το πολύ τριπλάσιο από την βέλτιστη ελάχιστη λύση που έχει μέγεθος OPT.

Γενικότερα, ένας ϵ-προσεγγιστικός αλγόριθμος για ένα πρόβλημα ελαχιστοποίησης βρίσκει μια λύση που έχει μέγεθος μικρότερο ή ίσο από $(1 + \epsilon)$ της βέλτιστης ελάχιστης λύσης OPT. Πράγματι,

$$\frac{a - \mathrm{OPT}}{\mathrm{OPT}} \leq e \Leftrightarrow a - \mathrm{OPT} \leq e \cdot \mathrm{OPT} \Leftrightarrow a \leq (1 + \epsilon)\mathrm{OPT}.$$

Αντίστοιχα, για προβλήματα μεγιστοποίησης ένας ϵ-προσεγγιστικός αλγόριθμος είναι ένας αλγόριθμος που βρίσκει μια λύση με μέγεθος μεγαλύτερο ή ίσο από $(1-\epsilon)$

της βέλτιστης μέγιστης λύσης OPT. Πράγματι, τότε ισχύει

$$\frac{\mathrm{OPT}-a}{\mathrm{OPT}} \leq e \Leftrightarrow \mathrm{OPT}-a \leq e \cdot \mathrm{OPT} \Leftrightarrow a \geq (1-\epsilon)\mathrm{OPT}.$$

Για παράδειγμα, ένας 1/3-προσεγγιστικός αλγόριθμος για ένα πρόβλημα μεγιστοποίησης βρίσκει μια λύση που έχει μέγεθος μεγαλύτερο ή ίσο από το $1-1/3=2/3$ της βέλτιστης μέγιστης λύσης OPT.

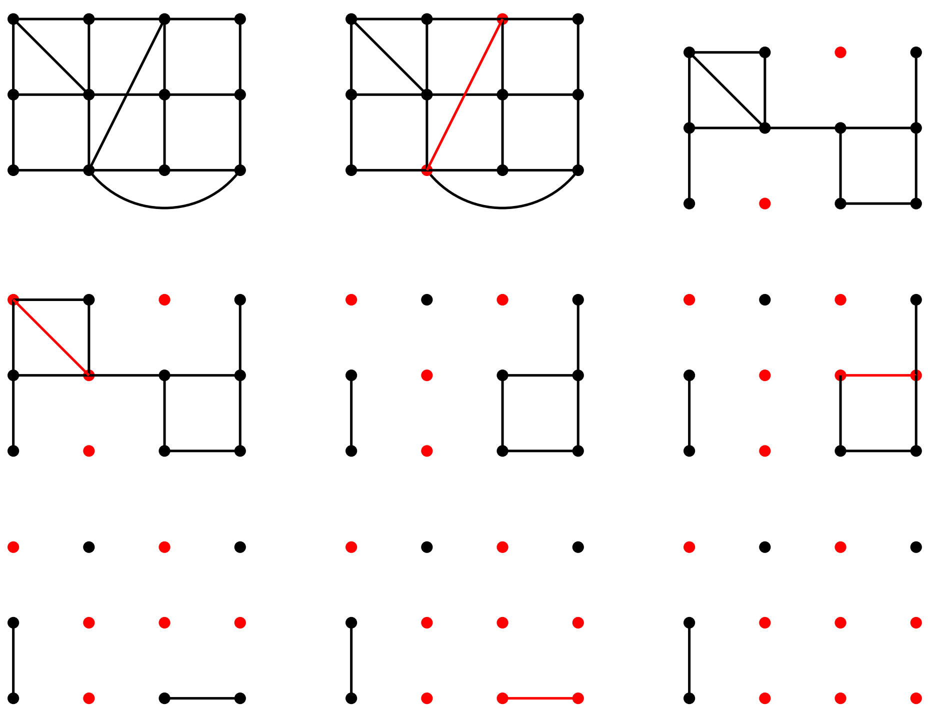
Στην συνέχεια θα δοθεί ένας 1-προσεγγιστικός αλγόριθμος για το πρόβλημα του καλύμματος από κορυφές του G, δηλαδή ένας αλγόριθμος που εξασφαλισμένα βρίσκει μια λύση με μέγεθος μικρότερο ή ίσο από το $(1{+}1)=2$-πλάσιο της βέλτιστης ελάχιστης λύσης:

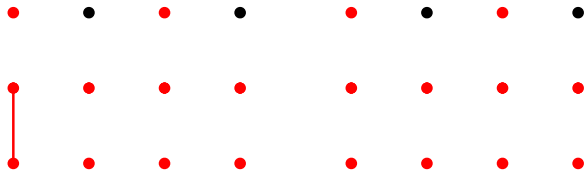
1-προσεγγιστικός αλγόριθμος για το VertexCover
Είσοδος: Ένα γράφημα δεσμών $G=(V,E)$
Έξοδος: Ένα κάλυμμα κορυφών B του G με $|B| \leq 2 \cdot \mathrm{OPT}$, όπου OPT το μέγεθος του ελάχιστου καλύμματος από κορυφές.

- Έστω B το ζητούμενο κάλυμμα από κορυφές. Αρχικά $B=\emptyset$.
- Ξεκινάμε από ένα τυχαίο δεσμό $e=\{u,v\}$ του G. Προσθέτουμε τα άκρα του e στο B και διαγράφουμε όλους τους δεσμούς του G που καλύπτονται από τις κορυφές u ή/και v.
  Στην συνέχεια, διαλέγουμε έναν από τους δεσμούς e' που απέμειναν μετά την διαγραφή. Προσθέτουμε και πάλι τα άκρα του e' στο B και διαγράφουμε όλους τους δεσμούς που καλύπτονται από τα άκρα αυτά.
  Συνεχίζουμε μέχρι να διαγράψουμε; όλους τους δεσμούς.
- Το τελικό σύνολο B είναι ένα κάλυμμα από κορυφές του G.

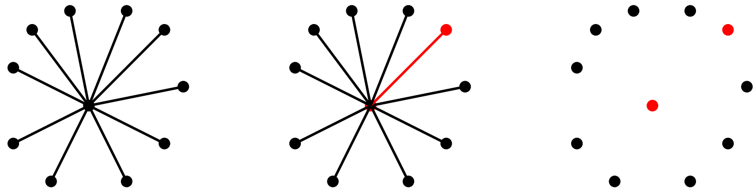
Παραδείγματα

(1)





(2)



Θα αποδείξουμε ότι αν B^* είναι ένα ελάχιστο κάλυμμα από κορυφές του G τότε για το σύνολο B που παράγει ο προηγούμενος αλγόριθμος ισχύει ότι $|B| \leq 2|B^*|$, δηλαδή ο αλγόριθμος είναι ένας 1-προσεγγιστικός αλγόριθμος για το πρόβλημα καλύμματος από κορυφές.

Έστω S το σύνολο των δεσμών που επιλέχθηκαν κατά την διαδικασία σχηματισμού του καλύμματος B. Παρατηρούμε ότι $|B| = 2|S|$. Οποιοδήποτε κάλυμμα από κορυφές του G πρέπει να έχει τουλάχιστον μια κορυφή που να καλύπτει τους δεσμούς του S. Επομένως και κάθε βέλτιστο ελάχιστο κάλυμμα B^* πρέπει να έχει αυτή την ιδιότητα. Επομένως, επειδή οι δεσμοί του S περιέχουν διαφορετικές κορυφές ο καθένας, έπεται ότι

$$|S| \leq |B^*| \Leftrightarrow \frac{|B|}{2} \leq |B^*| \Leftrightarrow |B| \leq 2|B^*| = (1+1)|B^*|$$

δηλαδή ο αλγόριθμος αυτός είναι ένας 1-προσεγγιστικός αλγόριθμος για την εύρεση του ελάχιστου καλύμματος από κορυφές

Η πολυπλοκότητα του αλγορίθμου είναι γραμμική ως προς τον αριθμό των κορυφών του γραφήματος. Παρ όλη την απλότητα του δεν υπάρχει άλλος καλύτερος (γραμμικός ή γρηγορότερος) γνωστός ϵ-προσεγγιστικός αλγόριθμος για το πρόβλημα του καλύμματος από κορυφές με μικρότερο ϵ.

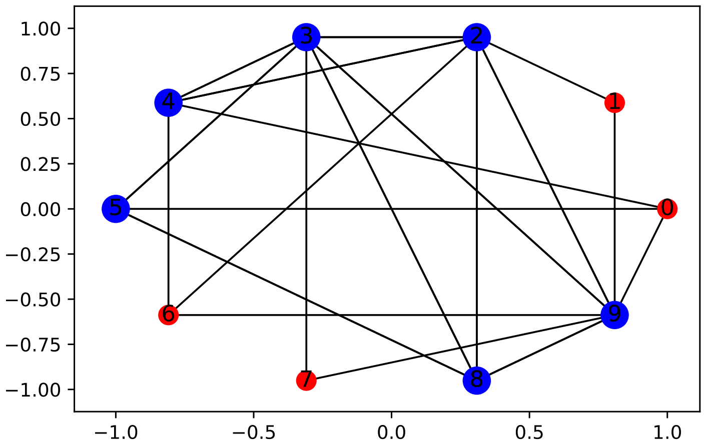
Η βιβλιοθήκη `networkx` διαθέτει την μέθοδο
`algorithms.approximation.min_weighted_vertex_cover(G, weight=None)`
η οποία υλοποιεί τον παραπάνω προσεγγιστικό αλγόριθμο. Μια απλοϊκή υλοποίηση της είναι η επόμενη:

```
import networkx as nx
import matplotlib.pyplot as plt
import random

G = nx.gnm_random_graph(10,20)

vertexcover = []
H = nx.Graph(G) #make a copy of G
while(len(H.edges) > 0):
    [u,v] = random.choice(list(H.edges()))
    vertexcover.append(u)
```

```
    vertexcover.append(v)
    H.remove_node(u)
    H.remove_node(v)
print(vertexcover)

pos = nx.layout.circular_layout(G)
nx.draw_networkx(G,pos,node_color='red', node_size=100.0)
G1 = G.subgraph(vertexcover)
nx.draw_networkx(G1,pos,node_color='blue', node_size=200.0)
plt.savefig("images/vertexcover01a.eps")
plt.show()
```

Output:

```
[2, 4, 3, 9, 5, 8]
```



Οι κορυφές με μπλε χρώμα αποτελούν ένα κάλυμμα από κορυφές των δεσμών του γραφήματος. Το βέλτιστο (ελάχιστο) κάλυμμα κορυφών περιέχει τουλάχιστον $6/2 = 3$ κορυφές.

Πιθανοτική απόδειξη. Η δεύτερη τεχνική ονομάζεται **πιθανοτική μέθοδος απόδειξης** (probabilistic proof method) και χρησιμοποιείται για να δείξουμε ότι κάποιο αντικείμενο υπάρχει χωρίς να χρειάζεται να το βρούμε ή να το κατασκευάσουμε. Ο θεμελιωτής της πιθανοτικής μεθόδου απόδειξης είναι ο Paul Erdős. Η απόδειξη με την πιθανοτική μέθοδο βασίζεται στην παρατήρηση ότι αν ένα ενδεχόμενο έχει θετική πιθανότητα να συμβεί τότε υπάρχει τουλάχιστον ένα στοιχείο του δειγματικού χώρου για το οποίο πραγματοποιείται αυτό το ενδεχόμενο.

Προκειμένου να εφαρμόσουμε αυτή την ιδέα σε ένα πρόβλημα αρχικά δημιουργούμε, με χρήση τυχαίων πειραμάτων, έναν κατάλληλο δειγματικό χώρο τα στοιχεία του οποίου αποτελούνται από αντικείμενα που μας ενδιαφέρουν και έπειτα αποδεικνύουμε ότι το αντικείμενο που μας ενδιαφέρει έχει θετική πιθανότητα να εμφανισθεί.

Στην επόμενη πρόταση παρουσιάζεται αναλυτικά η χρήση της πιθανοτικής μεθόδου στο πρόβλημα του ανεξάρτητου συνόλου. Το 1941 ο Turan απέδειξε ότι αν ένα γράφημα έχει σχετικά μικρό αριθμό δεσμών τότε διαθέτει ένα αρκετά σχετικά μεγάλο ανεξάρτητο σύνολο κορυφών.

Πρόταση 40 (Turan, 1941). *Αν ένα γράφημα έχει n κορυφές και m δεσμούς, όπου $\frac{n}{2} \leq m \leq \frac{n^2}{4}$, τότε περιέχει ένα ανεξάρτητο σύνολο με μέγεθος τουλάχιστον $\frac{n^2}{4m}$.*

Απόδειξη. Θα δοθεί μια πιθανοτική απόδειξη.

Αρχικά δημιουργούμε με χρήση τυχαίων πειραμάτων έναν κατάλληλο δειγματικό χώρο τα στοιχεία του οποίου αποτελούνται από αντικείμενα που μας ενδιαφέρουν. Στην απόδειξη αυτή μας ενδιαφέρουν σύνολα κορυφών του γραφήματος.

Θεωρούμε ως τυχαίο πείραμα την επιλογή ενός συνόλου S κορυφών του γραφήματος. Συγκεκριμένα, προκειμένου να κατασκευάσουμε το σύνολο S επιλέγουμε να συμπεριλάβουμε κάθε κορυφή με πιθανότητα p.

Πέρα από το πιθανοτικό κομμάτι, η απόδειξη χρησιμοποιεί την επόμενη σχεδόν "μαγική" προσέγγιση. Επειδή δεν μπορούμε να αποδείξουμε απευθείας την ύπαρξη ενός ανεξάρτητου συνόλου κορυφών, αντί αυτού θα αποδείξουμε ότι υπάρχει ένα υποσύνολο S των κορυφών του G το οποίο περιέχει τουλάχιστον $\frac{n^2}{4m}$ παραπάνω κορυφές από ότι δεσμούς. Στην περίπτωση αυτή, αν διαγράψουμε από το S την μία από τις δύο κορυφές κάθε δεσμού που περιέχει, τότε θα προκύψει ένα κορυφών σύνολο S' με τουλάχιστον $\frac{n^2}{4m}$ κορυφές το οποίο δεν θα περιέχει κανένα δεσμό, άρα το S' θα είναι ένα ανεξάρτητο σύνολο με μέγεθος τουλάχιστον $\frac{n^2}{4m}$.

Θεωρούμε τις τυχαίες μεταβλητές X και Y που εκφράζουν αντίστοιχα το πλήθος των κορυφών και των δεσμών του S.

Για κάθε κορυφή v_i του G ορίζουμε την τυχαία μεταβλητή

$$X_i = \begin{cases} 1, & \text{αν η κορυφή } v_i \text{ ανήκει στο } S \\ 0, & \text{αλλιώς.} \end{cases}$$

Η X_i είναι δείκτρια της ιδιότητας v_i ανήκει στο S.

Επίσης, για κάθε δεσμό e_j ορίζουμε την τυχαία μεταβλητή

$$Y_j = \begin{cases} 1, & \text{αν τα άκρα του } e_j \text{ ανήκουν στο } S \\ 0, & \text{αλλιώς.} \end{cases}.$$

Η Y_j είναι δείκτρια της ιδιότητας e_j ανήκει στο S.

Τότε

$$X = X_1 + X_2 + \cdots + X_n$$

και

$$Y = Y_1 + Y_2 + \cdots + Y_m$$

Επειδή οι τυχαίες μεταβλητές X_i και Y_j είναι δείκτριες τυχαίες μεταβλητές που λαμβάνουν μόνο τις τιμές 0 και 1 έπεται ότι η μέση τιμή τους ισούται με την πιθανότητα να πραγματοποιηθεί η επιθυμητή ιδιότητα που εκφράζουν. Συγκεκριμένα, κάθε κορυφή v_i ανήκει στο S πιθανότητα p, οπότε

$$E(X_i) = 1 \cdot p + 0 \cdot (1 - p) = p.$$

Επίσης, κάθε δεσμός e_j του γραφήματος ανήκει στο S με πιθανότητα p^2, οπότε

$$E(Y_j) = 1 \cdot p^2 + 0 \cdot (1 - p^2) = p^2.$$

Άρα, από την γραμμικότητα της μέσης τιμής προκύπτει ότι

$$E(X) = E(X_1 + X_2 + \cdots + X_n) = E(X_1) + E(X_2) + \cdots + E(X_n) = np$$

και

$$E(Y) = E(Y_1 + Y_2 + \cdots + Y_m) = E(Y_1) + E(Y_2) + \cdots + E(Y_m) = mp^2.$$

Όπως αναφέραμε στην αρχή της απόδειξης μας ενδιαφέρουν σύνολα κορυφών τα οποία έχουν όσο το δυνατόν περισσότερες κορυφές από ότι δεσμούς. Ένα επιχείρημα για αυτή την επιλογή είναι ότι κάθε ανεξάρτητο σύνολο κορυφών του G έχει την ιδιότητα ότι η διαφορά του πλήθους των κορυφών του και του πλήθους των δεσμών του είναι η μέγιστη δυνατή σε σχέση με το μέγεθός του.

Θεωρούμε την τυχαία μεταβλητή $Z = X - Y$. Πάλι από την γραμμικότητα της μέσης τιμής έχουμε ότι $E(Z) = E(X - Y) = E(X) - E(Y) = np - mp^2$.

Στο σημείο αυτό σκεφτόμαστε με τον επόμενο "απρόσμενο" τρόπο. Η μέση τιμή της Z γίνεται μέγιστη όταν

$$\frac{d}{dp}Z = 0 \Leftrightarrow \frac{d}{dp}(np - mp^2) = 0 \Leftrightarrow n - 2mp = 0 \Leftrightarrow p = \frac{n}{2m}.$$

Άρα, αν επιλέξουμε κάθε κορυφή του S με πιθανότητα $p = \dfrac{n}{2m}$ τότε η μέση τιμή $E(Z) = E(X - Y)$ γίνεται μέγιστη και ισούται με

$$E(Z) = n\frac{n}{2m} - m\frac{n^2}{4m^2} = \frac{2n^2 - n^2}{4m} = \frac{n^2}{4m}.$$

Επειδή η μέση τιμή της τυχαίας μεταβλητής $Z = X - Y$ ισούται με $\dfrac{n^2}{4m}$ έπεται ότι υπάρχει σύνολο κορυφών S του G για το οποίο η διαφορά $X - Y$ είναι μεγαλύτερη ή ίση από $\dfrac{n^2}{4m}$. Επομένως, σύμφωνα με την αρχική παρατήρηση, υπάρχει ένα ανεξάρτητο σύνολο κορυφών του G με μέγεθος τουλάχιστον $\dfrac{n^2}{4m}$. □

Έτσι, για παράδειγμα, σε ένα γράφημα με $n = 100$ κορυφές και $m = 125$ δεσμούς υπάρχει τουλάχιστον ένα ανεξάρτητο σύνολο με $\frac{n^2}{4m} = \frac{100^2}{4 \cdot 125} = \frac{10000}{500} = 20$ κορυφές.

Η πιθανοτική μέθοδος που ακολουθήσαμε στην προηγούμενη απόδειξη μπορεί να συνοψιστεί από τα επόμενα βήματα:

(1) Δημιουργούμε με χρήση τυχαίων πειραμάτων έναν κατάλληλο δειγματικό χώρο αντικειμένων που μας ενδιαφέρουν.

(2) Ορίζουμε μια τυχαία μεταβλητή X, η οποία αντιστοιχεί στα ποσοτικά χαρακτηριστικά (τον αριθμό ή το μέγεθος) των αντικειμένων που πληρούν την επιθυμητή ιδιότητα.

(3) Εκφράζουμε την τυχαία μεταβλητή X ως άθροισμα τυχαίων μεταβλητών

$$X = X_1 + X_2 + \cdots + X_n$$

όπου οι X_i, $1 \leq i \leq n$ είναι τυχαίες μεταβλητές της μορφής

$$X_i = \begin{cases} 1 & \text{αν ισχύει η επιθυμητή ιδιότητα} \\ 0 & \text{αλλιώς} \end{cases}$$

(4) Υπολογίζουμε την μέση τιμή $E(X_i)$ κάθε τυχαίας μεταβλητής X_i.

(5) Υπολογίζουμε την μέση τιμή $E(X)$ της X βάσει της ιδιότητας της γραμμικότητας της μέσης τιμής:

$$E(X) = E(X_1) + E(X_2) + \cdots + E(X_n)$$

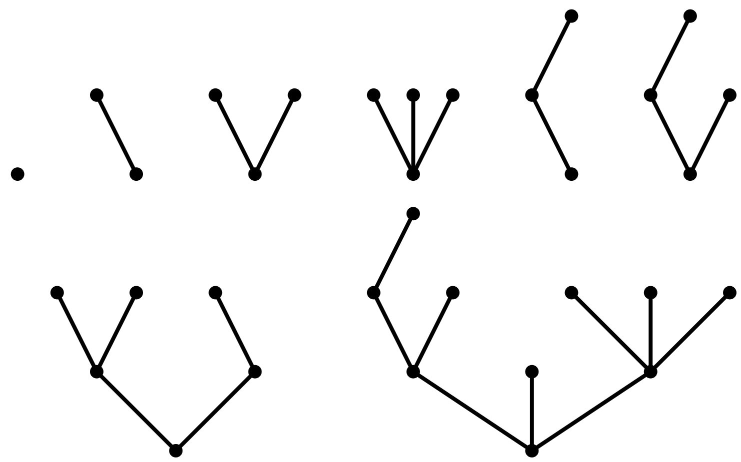
(6) Η απόδειξη της ύπαρξης ολοκληρώνεται με την παρατήρηση ότι μια οποιαδήποτε τυχαία μεταβλητή παίρνει τουλάχιστον μια τιμή μεγαλύτερη ή ίση από την μέση τιμή της και τουλάχιστον μια τιμή μικρότερη ή ίση με την μέση τιμή της, δηλαδή υπάρχει τουλάχιστον ένα σημείο (αντικείμενο) του δειγματικού χώρου για το οποίο $X \geq E(X)$ και τουλάχιστον ένα σημείο (αντικείμενο) του δειγματικού χώρου όπου $X \leq E(X)$.

ΔΕΥΤΕΡΟ ΜΕΡΟΣ: ΔΕΝΔΡΑ - ΔΙΑΤΕΤΑΓΜΕΝΑ ΔΕΝΔΡΑ - ΔΥΑΔΙΚΑ ΔΕΝΔΡΑ

17. ΔΕΝΔΡΑ

Ένα συνεκτικό άκυκλο γράφημα δεσμών λέγεται **δένδρο** (tree), ή **ελεύθερο δένδρο**).

Παραδείγματα :



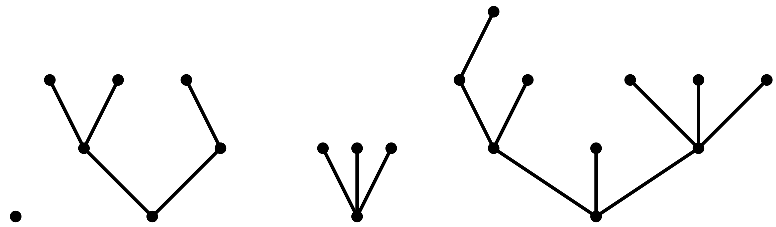
Οι κόμβοι ενός δένδρου με βαθμό 1 λέγονται **φύλλα** (leaves) του δένδρου. Κάθε μη τετριμμένο δένδρο έχει δύο τουλάχιστον φύλλα.

Δάση

Ένα άκυκλο γράφημα δεσμών λέγεται **δάσος** (forest).

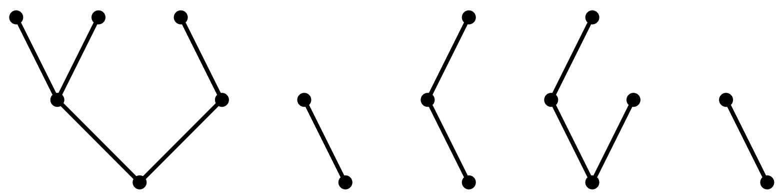
Κάθε δάσος είναι η ένωση δένδρων.

Παραδείγματα :



$\mathbf{T_1}$ $\mathbf{T_2}$ $\mathbf{T_3}$ $\mathbf{T_4}$

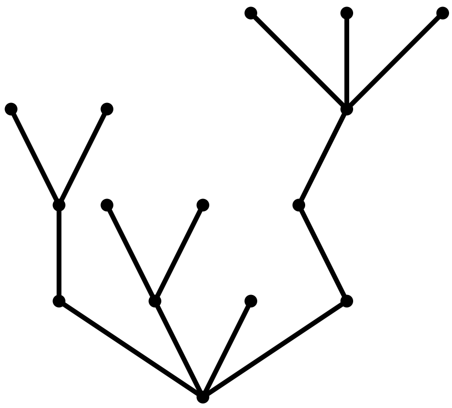
Το δάσος αυτό αποτελείται από την ένωση των δένδρων T_1, T_2, T_3, T_4.



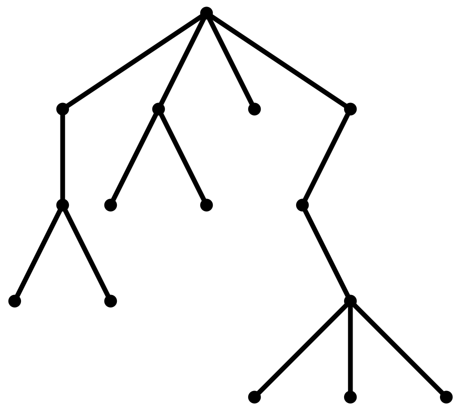
$\mathbf{T_1}$ $\mathbf{T_2}$ $\mathbf{T_3}$ $\mathbf{T_4}$ $\mathbf{T_5}$

Το δάσος αυτό αποτελείται από την ένωση των δένδρων T_1, T_2, T_3, T_4, T_5.

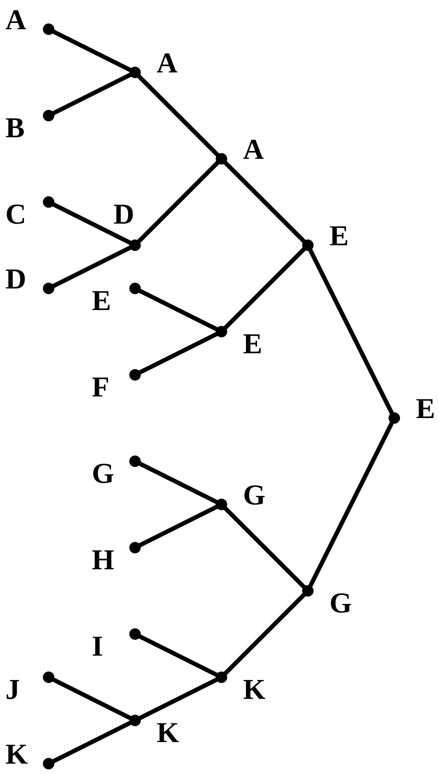
ΑΝΑΠΑΡΑΣΤΑΣΗ ΔΕΝΔΡΩΝ



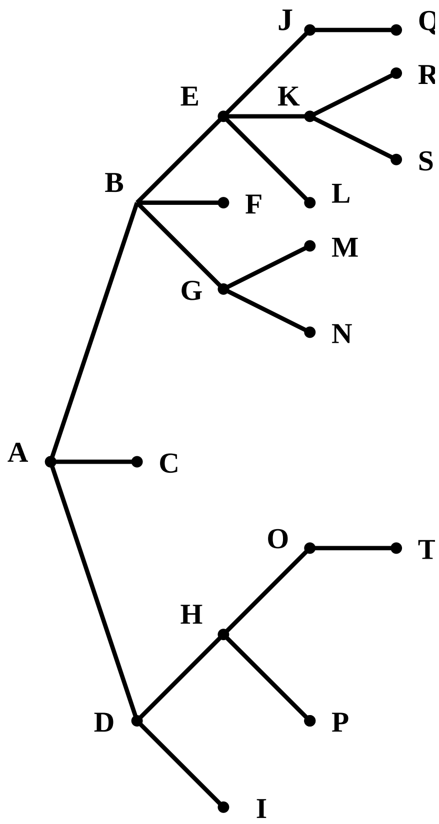
“Φυσικός” τρόπος



Στην πληροφορική



Τουρνουά



Γενεαλογικά δένδρα

Πρόταση 41. *Οι παρακάτω προτάσεις είναι ισοδύναμες:*

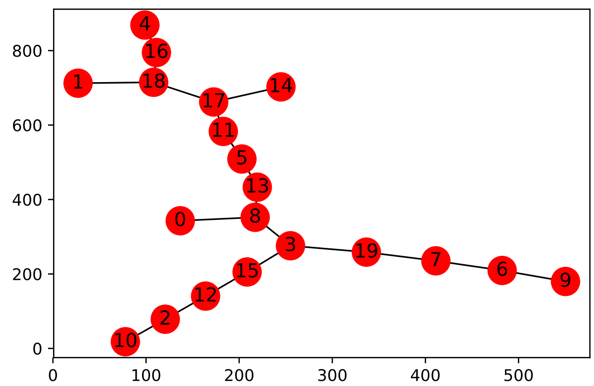
(i) Το γράφημα G είναι δένδρο.
(ii) Κάθε δύο κόμβοι του G ενώνονται με ένα μοναδικό μονοπάτι.
(iii) Το G είναι συνεκτικό και $|V| = |E| + 1$.
(iv) Το G είναι άκυκλο και $|V| = |E| + 1$.
(v) Το G είναι άκυκλο και αν ενώσουμε δύο οποιουσδήποτε μη γειτονικούς κόμβους του δημιουργούμε γράφημα με ακριβώς ένα κύκλο.
(vi) Το G είναι συνεκτικό και αν διαγράψουμε οποιονδήποτε δεσμό του δημιουργούμε μη συνεκτικό γράφημα.

Παρατήρηση: Σε κάθε δένδρο $T = (V, E)$ ισχύει ότι ο αριθμός $|V|$ των κορυφών του είναι ένα παραπάνω από τον αριθμό $|E|$ των δεσμών του. Δηλαδή, όλα τα δένδρα με $|V|$ κορυφές έχουν ακριβώς $|V| - 1$ δεσμούς.

Μάλιστα, λόγω της ιδιότητας (vi), τα δένδρα περιέχουν τον ελάχιστο αριθμό από δεσμούς που απαιτούνται για να είναι ένα γράφημα συνεκτικό. Με άλλα λόγια, ένα γράφημα με n κορυφές και λιγότερους από $n - 1$ δεσμούς είναι σίγουρα μη συνεκτικό.

Μπορούμε να χρησιμοποιήσουμε την βιβλιοθήκη networkx για να κατασκευάσουμε ένα τυχαίο δένδρο με n κορυφές.

```
import networkx as nx
import matplotlib.pyplot as plt

n = 20
T = nx.random_tree(n)
pos = nx.layout.kamada_kawai_layout(T)
nx.draw_networkx(T,pos)

plt.show()
```



Λόγω της ιδιότητας (i) τα περισσότερα αλγοριθμικά προβλήματα απλοποιούνται όταν τα γραφήματα που εξετάζουμε είναι δένδρα ή δάση.

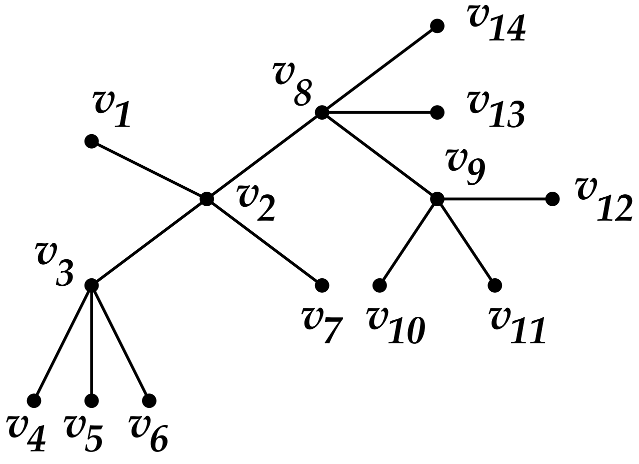
Πρόταση 42.

(i) Κάθε δάσος είναι διμερές γράφημα.
(ii) Κάθε δάσος έχει χρωματικό αριθμό 2.
(iii) Κάθε δάσος είναι επίπεδο γράφημα.
(iv) Τα μπλοκ ενός δάσους είναι οι δεσμοί του.

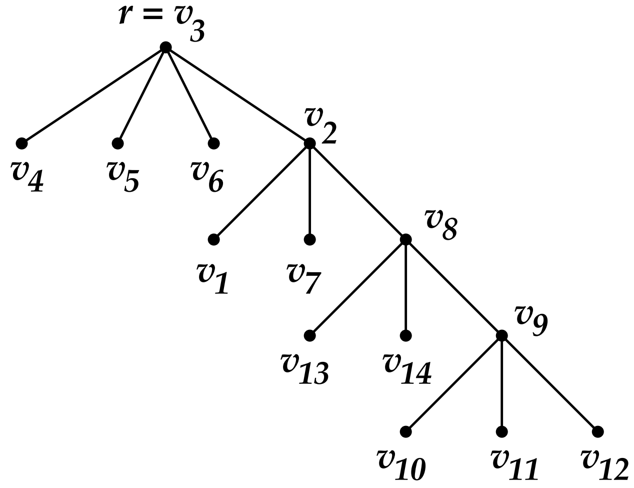
Στα επόμενα θα χρησιμοποιούμε το σύμβολο T αντί του G για τα γραφήματα που είναι δένδρα.

18. ΔΕΝΔΡΑ ΜΕ ΡΙΖΑ

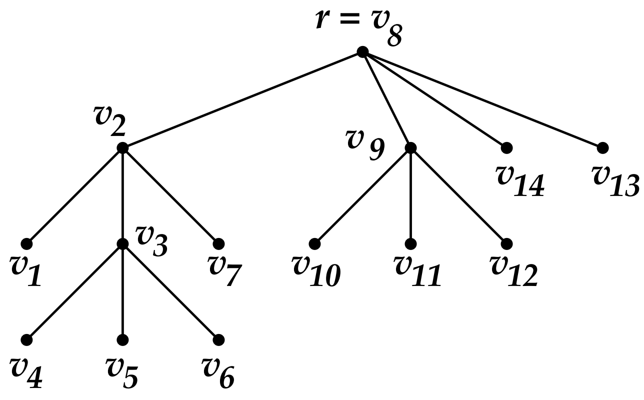
Δένδρο με ρίζα (rooted tree) είναι ένα δένδρο με έναν ειδικά επιλεγμένο κόμβο (τη **ρίζα** (root) του δένδρου).



Το δένδρο T



Το δένδρο T με ρίζα $r = v_3$

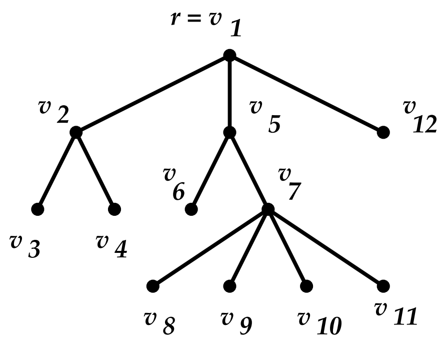


Το δένδρο T με ρίζα $r = v_8$

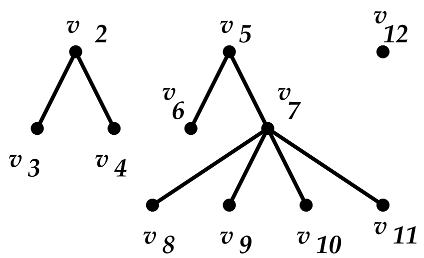
Παρατήρηση: Για να είναι δύο δένδρα με ρίζα ισόμορφα, πρέπει προφανώς να ισχύει ο ορισμός του ισομορφισμού των δένδρων, με την επιπλέον απαίτηση ότι η ρίζα του πρώτου δένδρου να απεικονίζεται στη ρίζα του δεύτερου.

Έστω r η ρίζα του δένδρου T. Τα δένδρα του δάσους $T - r$ λέγονται **υποδένδρα της ρίζας** r (subtrees). Τα δένδρα του δάσους $T - r$ θεωρούνται επίσης δένδρα με ρίζα : Ρίζα καθενός είναι το άλλο άκρο του δεσμού που περιέχει την ρίζα r του T. Η έννοια των υποδένδρων ενός οποιουδήποτε κόμβου (που δεν είναι φύλλο) ορίζεται ανάλογα.

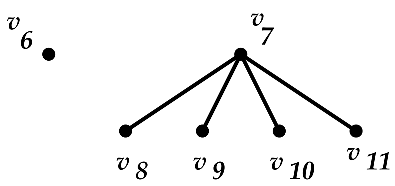
Παραδείγματα :



Υποδένδρα της ρίζας v_1 είναι τα



Υποδένδρα του κόμβου v_5 είναι τα



Ορίζουμε το **επίπεδο** (level) $l(v)$ ενός κόμβου v του T ως εξής: $l(r) = 0$ και αν στο (μοναδικό) $r - v$ μονοπάτι $(r, \ldots, u, v)$ έχουμε $l(u) = i$, τότε $l(v) = i + 1$. (Στην περίπτωση αυτή το u λέγεται **γονέας** (parent) του v και το v λέγεται **παιδί** (child) του u. Παιδιά του ίδιου γονέα λέγονται **αδέλφια** (siblings). Με άλλα λόγια το επίπεδο ενός κόμβου είναι η απόσταση του από την ρίζα του δένδρου.

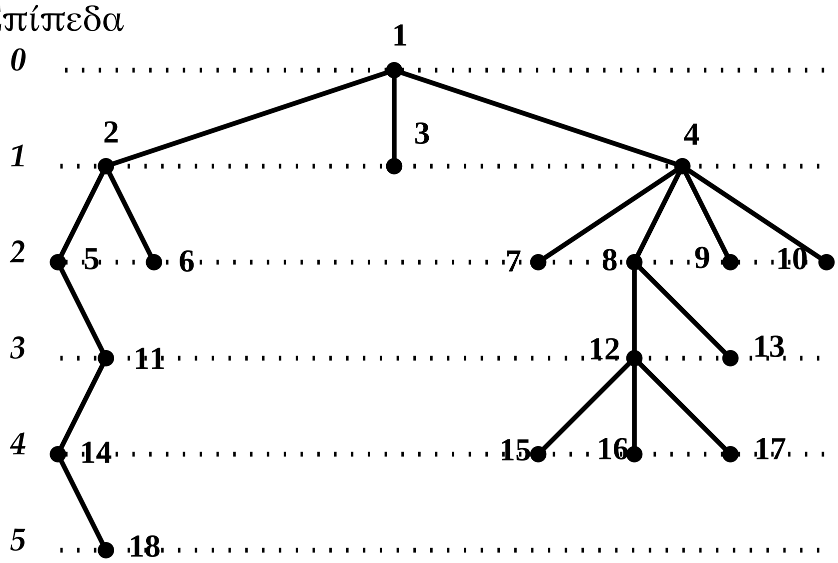
Το πλήθος των παιδιών ενός κόμβου σε ένα δένδρο με ρίζα ονομάζεται **βαθμός** του κόμβου, (οπότε, προφανώς, ο βαθμός κάθε κόμβου, εκτός από τη ρίζα, είναι κατά ένα μικρότερος από το βαθμό του κόμβου, όπως αυτός ορίσθηκε νωρίτερα για ένα τυχαίο γράφημα).

Αν υπάρχει στο T διαδρομή από ένα κόμβο v_1 σε ένα κόμβο v_k, η οποία χρησιμοποιεί κόμβους με επίπεδα που συνεχώς αυξάνουν τότε λέμε ότι το v_1 είναι **πρόγονος** (ancestor) του v_k και το v_k είναι **απόγονος** (descentant) του v_1.

Ένας κόμβος χωρίς παιδιά (δηλαδή βαθμού 0) λέγεται **φύλλο** (leaf) (ή **τερματικός κόμβος** (terminal node)). Αλλιώς λέγεται **ενδιάμεσος** κόμβος (internal node).

Ύψος (height) (ή **βάθος** (depth)) $h(T)$ ενός δένδρου T λέγεται το μεγαλύτερο από τα επίπεδα των κόμβων του.

Παράδειγμα: Επίπεδα

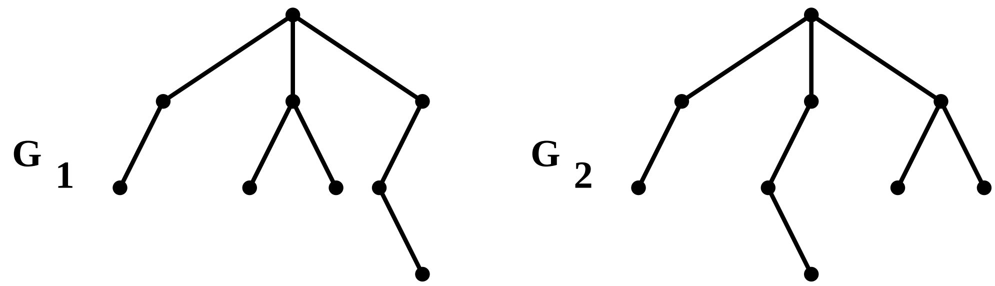


1 : ρίζα	2 : γονέας των 5, 6
5, 6 : παιδιά του 2	4 : γονέας του 7
7, 8, 9, 10 : παιδιά του 4	15, 16, 17 : αδέλφια
2 : πρόγονος των 5,6, 11,14,18	15 : απόγονος των 1, 4, 8, 12
3, 6, 7, 9, 10, 13, 15, 16, 17, 18 : φύλλα	Βαθμός του 12 = 3
11 : ενδιάμεσος κόμβος	$h(T) = 5$.

19. ΔΙΑΤΕΤΑΓΜΕΝΑ ΔΕΝΔΡΑ

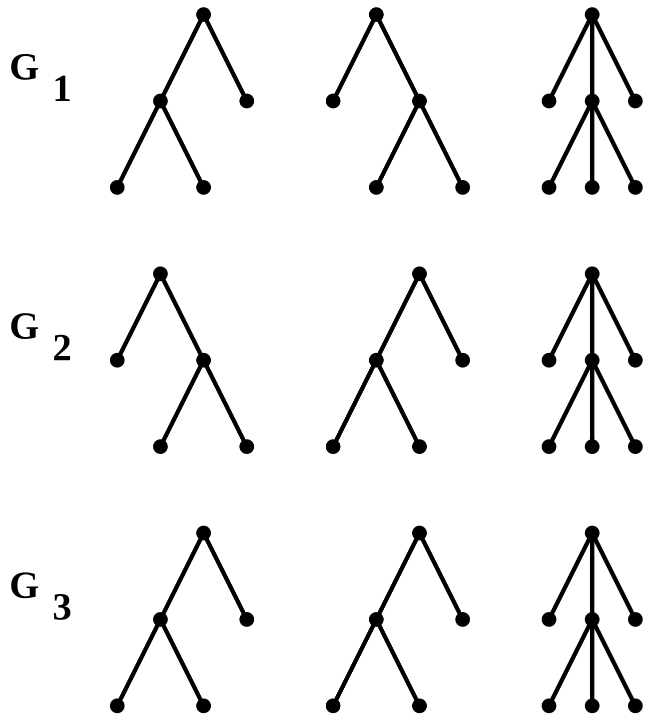
Ένα δένδρο με ρίζα λέγεται **διατεταγμένο** (ordered tree) αν η αλλαγή της σχετικής θέσης των υποδένδρων της ρίζας του θεωρείται ότι δημιουργεί μη ισόμορφο δένδρο.

Παράδειγμα:



Αν τα G_1, G_2 δεν θεωρηθούν διατεταγμένα τότε $G_1 \simeq G_2$, αλλά τα διατεταγμένα δένδρα G_1, G_2 δεν είναι ισόμορφα. (Σ΄ ένα διατεταγμένο δένδρο, τα υποδένδρα της ρίζας χαρακτηρίζονται σαν πρώτο, δεύτερο κ.λπ. από αριστερά πρός τα δεξιά).

Διατεταγμένο δάσος διατεταγμένων δένδρων είναι ένα διατεταγμένο σύνολο από ξένα, διατεταγμένα δέντρα.



Τα τρία δάση G_1, G_2, G_3 είναι ισόμορφα. Δεν είναι όμως ισόμορφα, αν θεωρηθούν ως διατεταγμένα δάση.

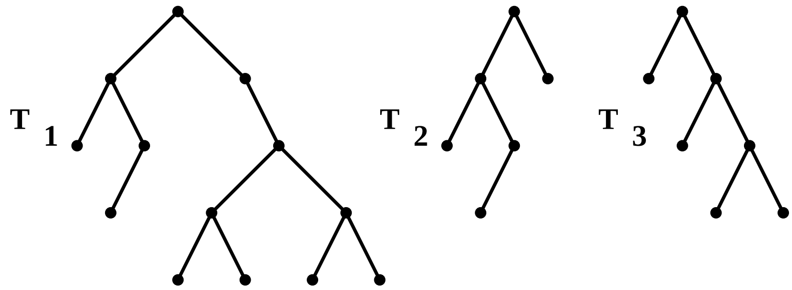
Ένα διατεταγμένο δένδρο, στο οποίο κάθε κόμβος επιτρέπεται να έχει το πολύ k παιδιά, λέγεται k-**δένδρο**.

Παράδειγμα:

Το πρώτο και το δεύτερο διατεταγμένο δένδρο του διατεταγμένου δάσους G_1 του τελευταίου παραδείγματος, είναι 2-δένδρα, ενώ το τρίτο είναι 3-δένδρο.

20. ΔΥΑΔΙΚΑ ΔΕΝΔΡΑ

Ένα δένδρο με ρίζα λέγεται **δυαδικό δένδρο** (binary tree) αν κάθε κόμβος του που δεν είναι φύλλο έχει είτε ένα αριστερό, είτε ένα δεξιό παιδί, είτε δύο παιδιά (ένα αριστερό και ένα δεξιό).

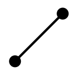


Τρία δυαδικά δένδρα T_1, T_2 και T_3.

Παρατήρηση:

(1) Σε αντίθεση με τον γενικό ορισμό των γραφημάτων, στα δυαδικά δένδρα συμπεριλαμβάνεται και το κενό δυαδικό δένδρο, δηλαδή το δένδρο T με $X(T) = \emptyset$.

(2) Τα δυαδικά δένδρα είναι διαφορετικά από τα διατεταγμένα 2-δένδρα. Για παράδειγμα, υπάρχει μόνο ένα διατεταγμένο 2-δένδρο με 2 κόμβους:

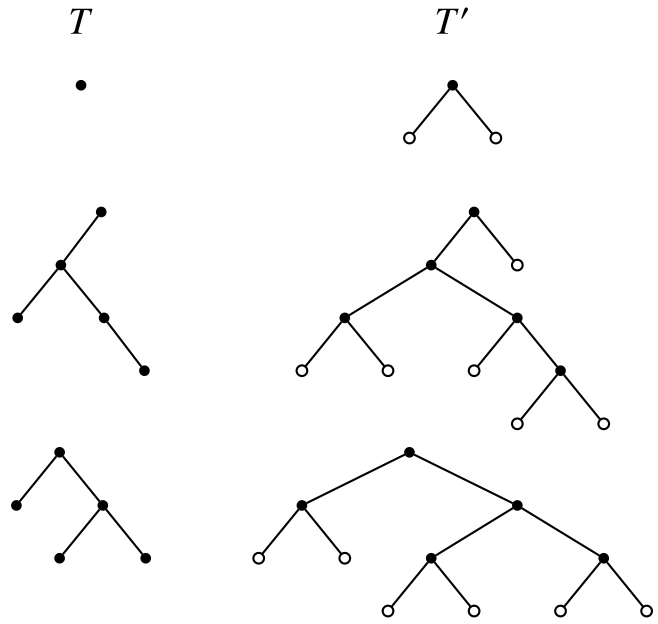


ενώ υπάρχουν δύο δυαδικά δένδρα με δύο κόμβους:

(3) Ισοδύναμος ορισμός του δυαδικού δένδρου μπορεί να δοθεί αναδρομικά:
- Το κενό δένδρο είναι δυαδικό.
- Κάθε μη κενό δυαδικό δένδρο T αποτελείται από ένα κόμβο r που έχει ως παιδιά δύο δυαδικά δένδρα T_1 (το αριστερό), T_2 (το δεξιό) τα οποία μπορεί να είναι κενά.

Οι δύο ορισμοί είναι ισοδύναμοι. Η ισοδυναμία είναι η εξής: Κάθε δυαδικό δένδρο T (πρώτος ορισμός) μπορεί να επεκταθεί κατά μοναδικό τρόπο σε ένα δυαδικό δένδρο T' (δεύτερος ορισμός), όπου κάθε κόμβος του έχει 0 ή 2 παιδιά, προσθέτοντας ένα αριστερό (αντ. δεξιό) (**κενό**) παιδί σε κάθε κόμβο με δεξιό (αντ. αριστερό) παιδί, και δύο (**κενά**) παιδιά σε κάθε φύλλο του T. Τα φύλλα του T' είναι όλα κενά δυαδικά δένδρα. Προφανώς, το δυαδικό δένδρο T μπορεί να προκύψει ξανά από το T' σβήνοντας όλα τα φύλλα του T'.

Παραδείγματα:



(4) Προφανώς, αντίστοιχα με το διατεταγμένο δάσος διατεταγμένων δένδρων ορίζεται και το **διατεταγμένο δάσος δυαδικών δένδρων**.

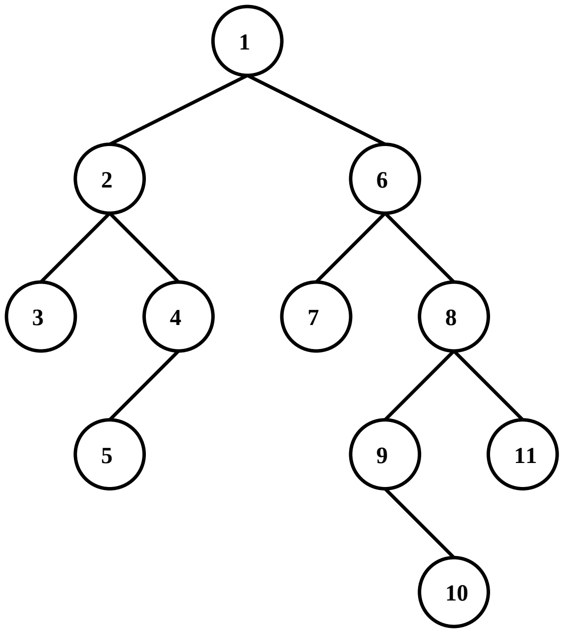
21. ΔΙΑΣΧΙΣΗ ΔΥΑΔΙΚΩΝ ΔΕΝΔΡΩΝ

Υπάρχουν 4 βασικοί τρόποι να διασχίσουμε (αριθμήσουμε) τους κόμβους ενός δυαδικού δένδρου.

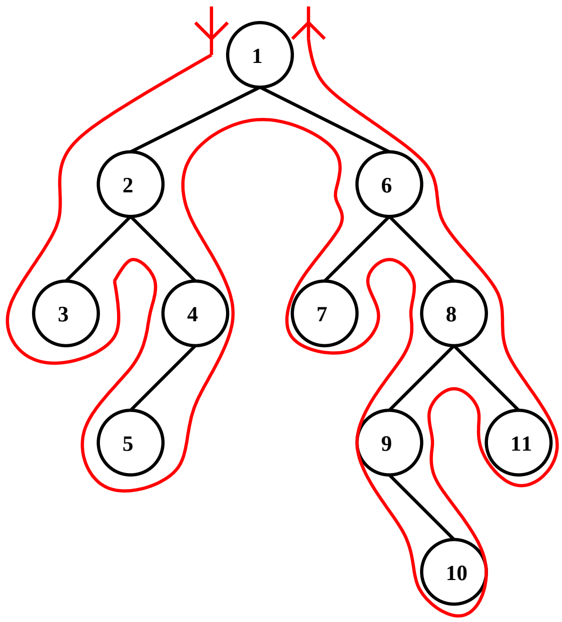
Καθένας απο τους τρόπους αυτούς καθορίζει μια ολική διάταξη των κόμβων του δένδρου.

1) Προδιάταξη : Επισκεπτόμαστε (αριθμούμε) κάθε κόμβο πρίν διασχίσουμε σε προδιάταξη το αριστερό και το δεξιό υποδένδρο του. Δηλαδή, επισκεπτόμαστε πρώτα τον γονέα και μετά τα δένδρα – παιδιά του (πρώτα το αριστερό και μετά το δεξιό).

Παράδειγμα:

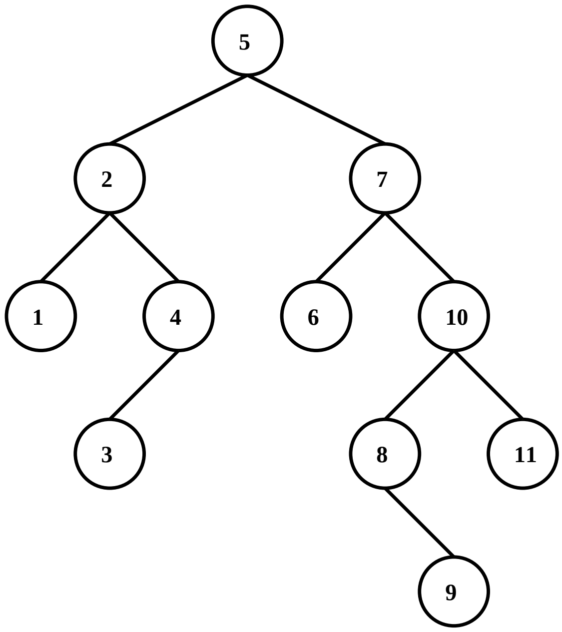


Πρακτικός τρόπος : Αριθμούμε κάθε κορυφή μόλις την πρωτοσυναντήσουμε καθώς κινούμαστε όπως δείχνει το σχήμα :

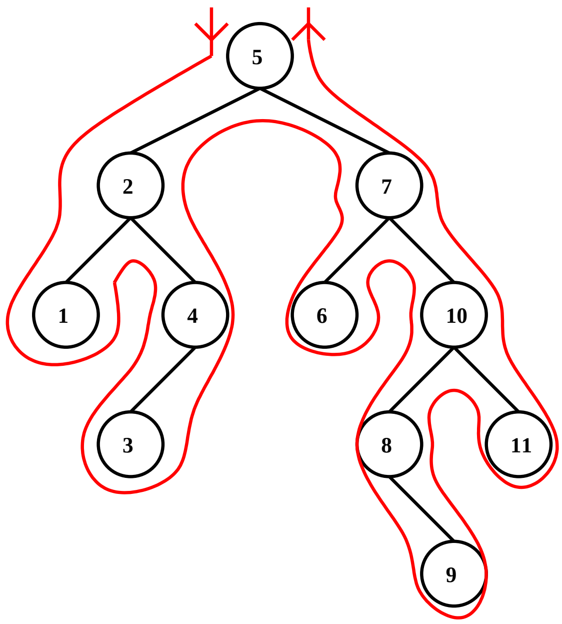


2) Ενδοδιάταξη : Επισκεπτόμαστε (αριθμούμε) κάθε κόμβο αφού διασχίσουμε σε ενδοδιάταξη το αριστερό υπόδενδρο του και πρίν την διασχίσουμε σε ενδοδιάταξη το δεξιό υποδένδρο του. Δηλαδή, επισκεπτόμαστε πρώτα το αριστερό δένδρο – παιδί, μετά τον γονέα και μετά το δεξιό δένδρο – παιδί.

Παράδειγμα:

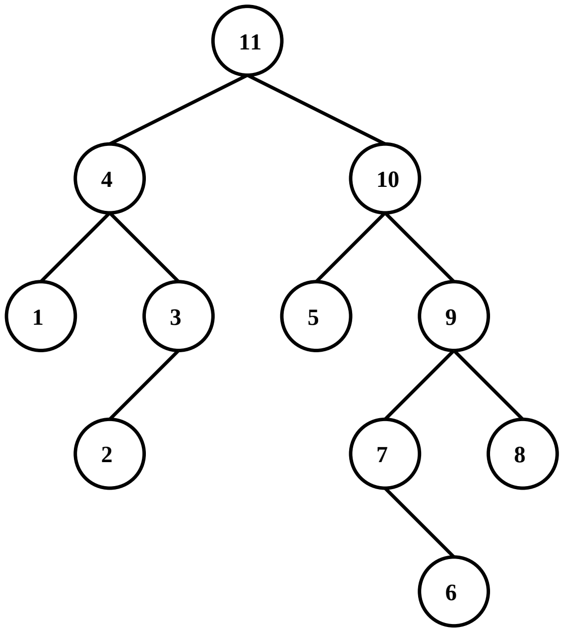


Πρακτικός τρόπος : Αριθμούμε κάθε κορυφή τη πρώτη φορά που τη συναντάμε αν δεν έχει αριστερό παιδί, ενώ την αριθμούμε τη δεύτερη φορά αν έχει αριστερό παιδί, καθώς κινούμαστε όπως δείχνει το σχήμα :

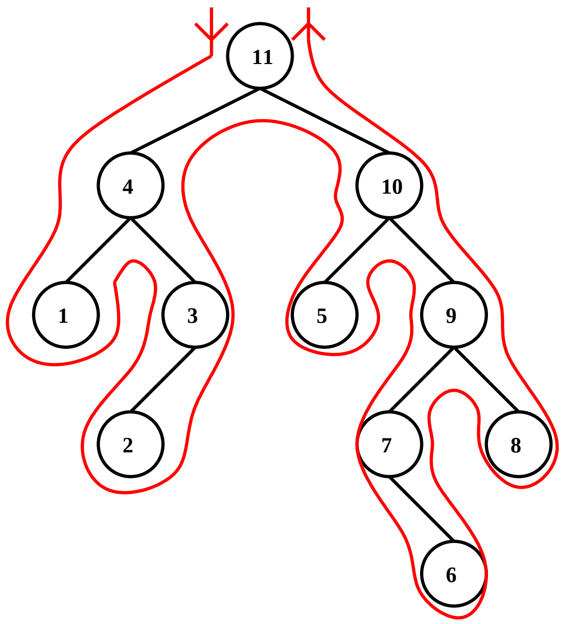


3) Μεταδιάταξη : Επισκεπτόμαστε (αριθμούμε) κάθε κόμβο αφού έχουμε διασχίσει σε μεταδιάταξη και το αριστερό και το δεξιό υποδένδρο του. Δηλαδή, επισκεπτόμαστε πρώτα τα δένδρα – παιδιά (πρώτα το αριστερό και μετά το δεξιό) και μετά τον γονέα.

Παράδειγμα:

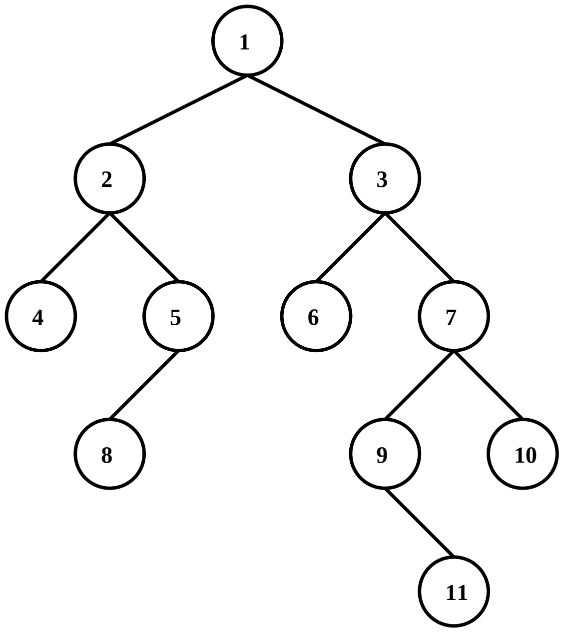


Πρακτικός τρόπος : Αριθμούμε κάθε κόμβο την τελευταία φορά που τον συναντάμε (δηλαδή καθως τον εγκαταλείπουμε για να πάμε προς τον γονέα του) καθώς κινούμαστε όπως δείχνει το σχήμα :



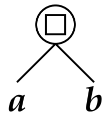
4) Διάσχιση κατά σειρά επιπέδων : Επισκεπτόμαστε (αριθμούμε) τους κόμβους κατά επίπεδο (από το μικρότερο προς το μεγαλύτερο), όπου σε κάθε επίπεδο επισκεπτόμαστε τους κόμβους από τα αριστερά προς τα δεξιά.

Παράδειγμα:

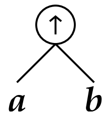


22. ΔΕΝΔΡΑ ΚΑΙ ΠΡΑΞΕΙΣ

Μια αλγεβρική παράσταση, στην οποία εμφανίζονται οι (δυαδικές) πράξεις $+$, $-$, $*$ (ή $\cdot$), $\div$ (ή $:$, ή $/$) καθώς και δυνάμεις, μπορεί να παρασταθεί σαν ένα δυαδικό δένδρο με ρίζα, αν γράψουμε



αντί για $a\square b$ (όπου $\square$ είναι οποιαδήποτε από τις τέσσερις πράξεις) και



αντί για a^b.

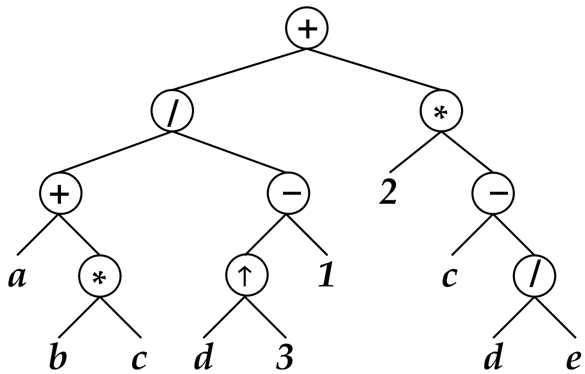
Γράφοντας τα σύμβολα με τη σειρά που εμφανίζονται στη διάσχιση του δένδρου σε προδιάταξη, σχηματίζουμε τη λεγόμενη **πολωνική** (ή **προθεματική**) **γραφή** της παράστασης.

Παραδείγματα:

1. Στην αλγεβρική παράσταση

$$\frac{a+bc}{d^3-1}+2(c-\frac{d}{e})$$

αντιστοιχεί το δυαδικό δένδρο



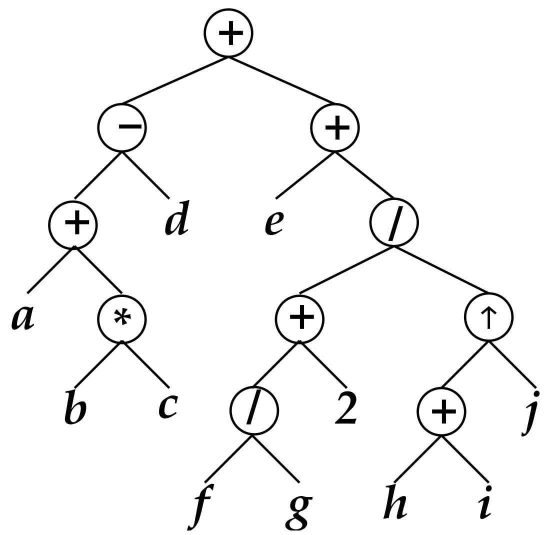
Άρα η πολωνική γραφή δίνει:

$$+/+a*bc-\uparrow d3\ \ 1*2-c/de.$$

2. Στην αλγεβρική παράσταση

$$((a + bc) - d) + \left(e + \frac{\frac{f}{g} + 2}{(h + i)^j}\right)$$

αντιστοιχεί το δυαδικό δένδρο



Άρα η πολωνική γραφή δίνει:

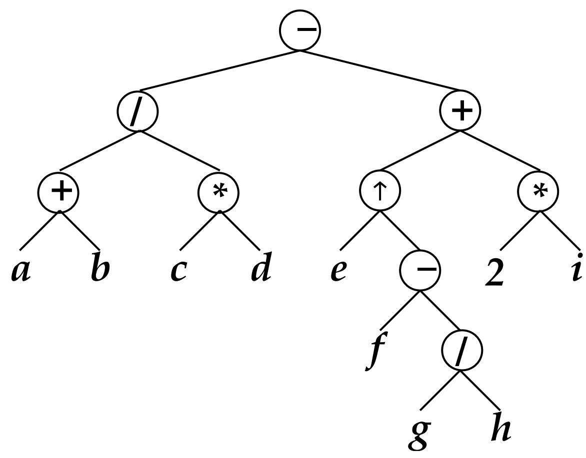
$$+ - +a * bcd + e/ + /fg2 \uparrow +hij.$$

Παρατήρηση: Προφανώς η αντιστοιχία ανάμεσα στους αλγεβρικούς τύπους και την πολωνική γραφή τους είναι αμφιμονοσήμαντη.

Παράδειγμα: Στην πολωνική γραφή

$$-/ + ab * cd + \uparrow e - f/gh * 2i,$$

αντιστοιχεί το δυαδικό δένδρο



και κατ΄ επέκταση η αλγεβρική παράσταση

$$\frac{a+b}{cd} - (e^{f - \frac{g}{h}} + 2i).$$

22.1. Πολυπλοκότητα αλγεβρικών παραστάσεων. Ένα μέτρο της πολυπλοκότητας (υπολογισμού) μιας αλγεβρικής παράστασης είναι ο λεγόμενος **αριθμός Stralher** ή register number που ορίζεται στο δυαδικό (ή k-αδικό) δένδρο που αναπαριστά την αλγεβρική παράσταση.

Συγκεκριμένα, σε κάθε κορυφή v ενός δυαδικού (ή k-αδικού) δένδρου T αντιστοιχεί ένας μοναδικός φυσικός αριθμός $s(v)$ που ονομάζεται αριθμός Strahler της κορυφής και ορίζεται ως εξής:

- Αν η κορυφή v είναι φύλλο τότε $s(v) = 1$.
- Αν η κορυφή v έχει ακριβώς ένα παιδί με αριθμό Strahler j και όλα τα άλλα παιδιά της έχουν μικρότερο αριθμό Stralher τότε $s(v) = j$.
- Αν η κορυφή v έχει δύο ή περισσότερα παιδιά με αριθμό Strahler j και όλα τα άλλα παιδιά της έχουν μικρότερο αριθμό Strahler τότε $s(v) = j + 1$.

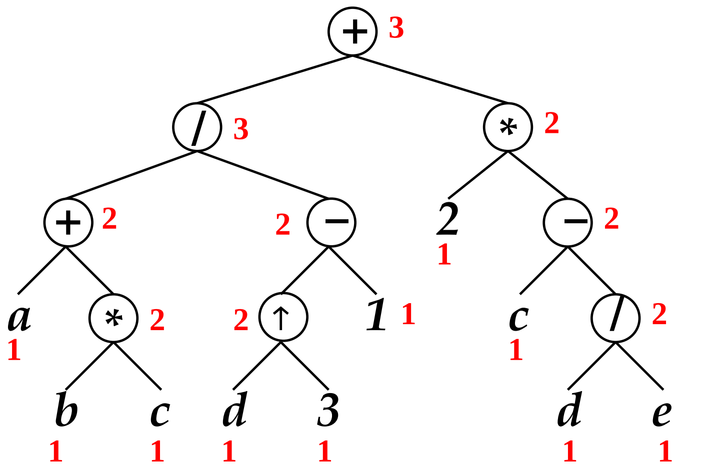
Ο αριθμός Strahler $s(T)$ του δένδρου T ορίζεται ως αριθμός Strahler της ρίζας του δένδρου. Αλγοριθμικά οι αριθμοί Strahler υπολογίζονται διασχίζοντας το δένδρο σε μεταδιάταξη.

Παραδείγματα:

Στο δυαδικό δένδρο που αντιστοιχεί στην αλγεβρική παράσταση

$$\frac{a + bc}{d^3 - 1} + 2(c - \frac{d}{e})$$

οι αριθμοί Strahler (σημειώνονται με κόκκινο) των κορυφών του είναι:

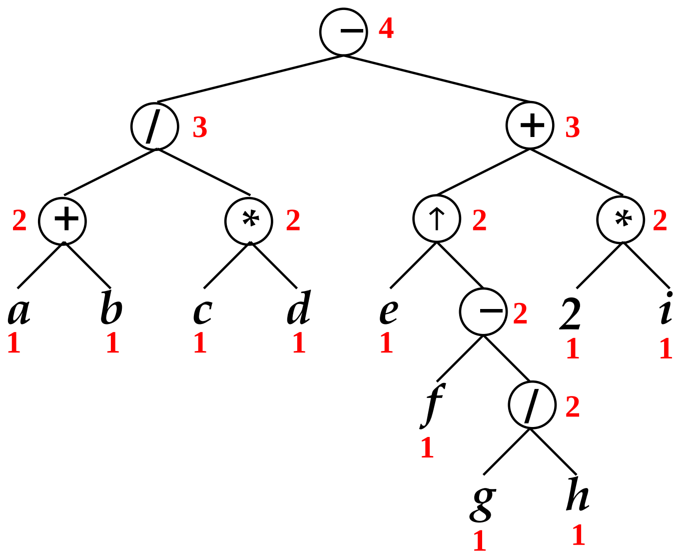


οπότε ο αριθμός Strahler της αλγεβρικής παράστασης είναι 3.

Στο δυαδικό δένδρο που αντιστοιχει στην αλγεβρική παράσταση

$$\frac{a + b}{cd} - (e^{f - \frac{g}{h}} + 2i).$$

οι αριθμοί Strahler (σημειώνονται με κόκκινο) των κορυφών του είναι:



οπότε ο αριθμός Strahler της αλγεβρικής παράστασης είναι 4.

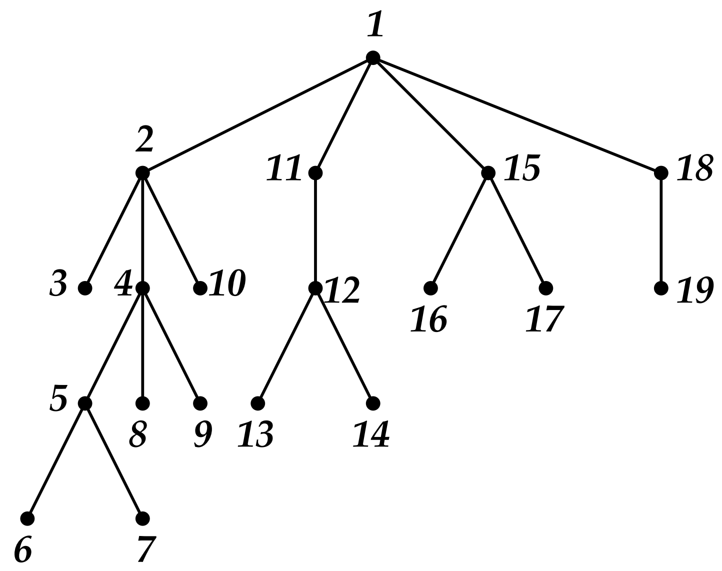
Ο αριθμός Strahler μιας αλγεβρικής παράστασης ισούται με τον ελάχιστο αριθμό καταχωρητών που απαιτούνται για τον υπολογισμό της αλγεβρικής παράστασης σε μια γλώσσα μηχανής. Η ιδέα είναι ότι σε κάθε κορυφή υπολογίζουμε πρώτα την (υπο) παράσταση του δένδρου-παιδιού της που έχει τον μεγαλύτερο αριθμό Strahler, μετά τα υπόλοιπα δένδρα-παιδιά της σε σειρά φθίνουσα σειρά ως προς τον αριθμο Stralher τους και έπειτα την πράξη που αντιστοιχεί στην κορυφή. (Βλέπε και τον αλγόριθμο των Sethi και Ullman).

23. ΔΙΑΣΧΙΔΗ ΔΙΑΤΕΤΑΓΜΕΝΩΝ ΔΕΝΔΡΩΝ

1) Προδιάταξη

Επισκεπτόμαστε (αριθμούμε) κάθε κόμβο πριν διασχίσουμε (σύμφωνα με τη διάταξή τους) τα δένδρα-παιδιά του σε προδιάταξη. Δηλαδή πρώτα τον γονέα και έπειτα τα δένδρα παιδιά του (από το πρώτο προς το τελευταίο).

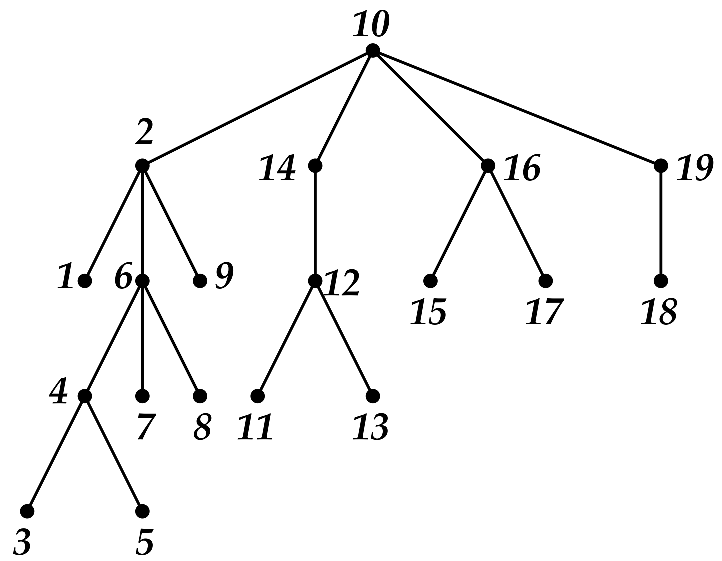
Παράδειγμα:



2) Ενδοδιάταξη

Επισκεπτόμαστε (αριθμούμε) κάθε κόμβο αφού διασχίσουμε σε ενδοδιάταξη το πρώτο δένδρο-παιδί και πριν διασχίσουμε (σύμφωνα με την διάταξή τους) τα υπόλοιπα δένδρα-παιδιά του σε ενδοδιάταξη. Δηλαδή πρώτα το πρώτο δένδρο-παιδί, μετά τον γονέα κι έπειτα τα υπόλοιπα δένδρα-παιδιά του (από το δεύτερο προς το τελευταίο).

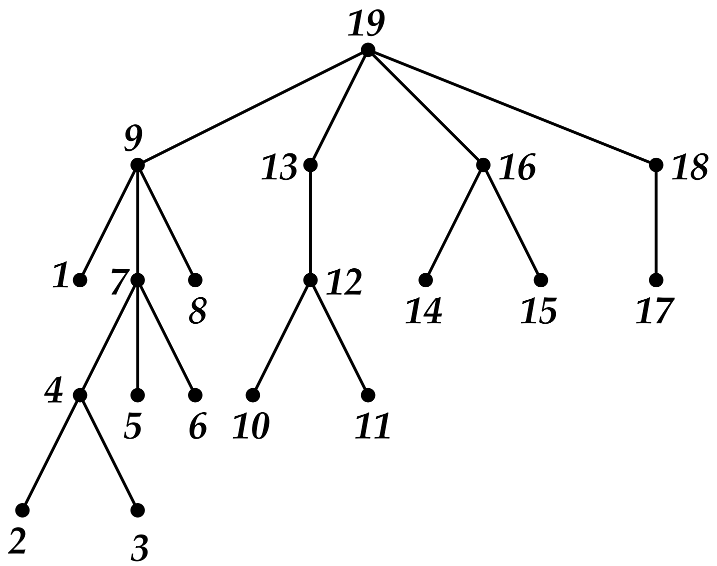
Παράδειγμα:



3) Μεταδιάταξη

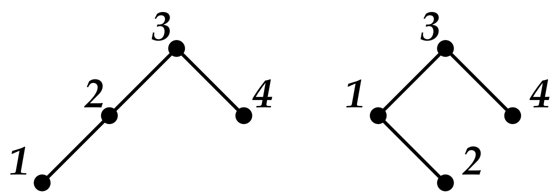
Επισκεπτόμαστε (αριθμούμε) κάθε κόμβο αφού διασχίσουμε (σύμφωνα με τη διάταξή τους) τα δένδρα-παιδιά του σε μεταδιάταξη. Δηλαδή πρώτα τα δένδρα-παιδιά (από το πρώτο προς το τελευταίο) και έπειτα τον γονέα.

Παράδειγμα:

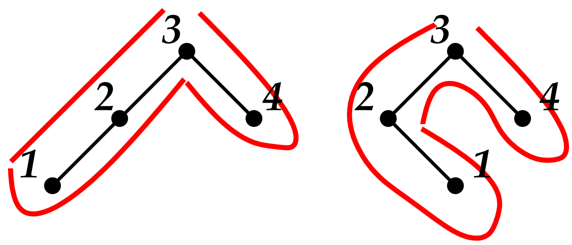


Παρατήρηση: Μπορούμε να αριθμήσουμε τα διατεταγμένα δένδρα σε προδιάταξη και μεταδιάταξη με πρακτικό τρόπο, αντίστοιχα με τα δυαδικά δένδρα. Ο πρακτικός τρόπος για την ενδοδιάταξη των διατεταγμένων δένδρων όμως είναι διαφορετικός: Αριθμούμε τον κάθε κόμβο τη δεύτερη φορά που τον συναντάμε, εκτός αν είναι φύλλο, οπότε τον αριθμούμε την πρώτη φορά. Η διαφορά αυτή οφείλεται στο ότι στα δυαδικά δένδρα, στην περίπτωση γονέα με μοναδικό παιδί η σειρά αρίθμησης τους δεν είναι μονοσήμαντα καθορισμένη. Αν το παιδί είναι αριστερό παιδί τότε προηγείται του γονέα ενώ αν είναι δεξιό τότε έπεται του γονέα.

Έτσι για παράδειγμα, ενώ η ενδοδιάταξη στα δύο παρακάτω δυαδικά δένδρα δίνει



ο παραπάνω πρακτικός τρόπος θα έδινε αντίστοιχα

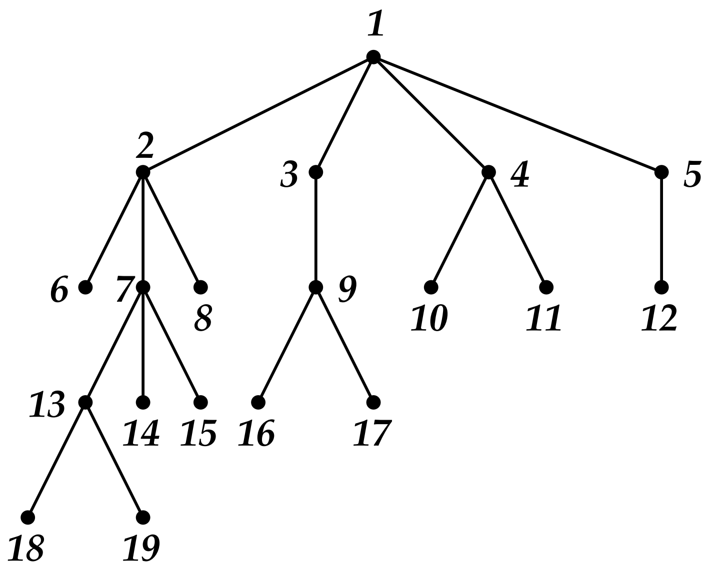


με το δεύτερο δένδρο να δίνει διαφορετική ενδοδιάταξη από ό,τι ο ορισμός.

4. Διάταξη κατά επίπεδα

Επισκεπτόμαστε (αριθμούμε) τους κόμβους κατά επίπεδο (από το μικρότερο επίπεδο στο μεγαλύτερο), όπου σε κάθε επίπεδο επισκεπτόμαστε τους κόμβους από τα αριστερά προς τα δεξιά.

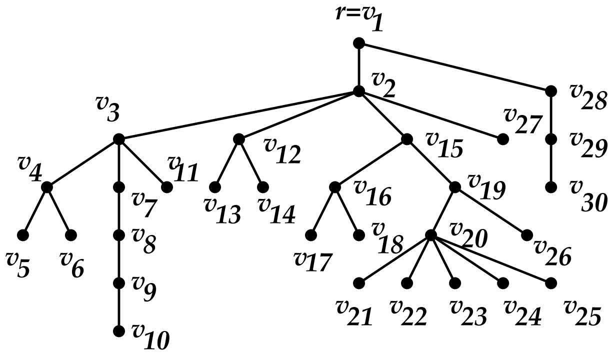
Παράδειγμα:



Παρατήρηση: Οι διασχίσεις των δένδρων (δυαδικών, ή διατεταγμένων) σύμφωνα με οποιαδήποτε από τις παραπάνω διατάξεις γενικεύονται προφανώς στα διατεταγμένα δάση, διασχίζοντας σύμφωνα με τη συγκριμένη κάθε φορά διάταξη το πρώτο δένδρο του διατεταγμένου δάσους, ακολούθως το δεύτερο δένδρο, κ.ο.κ.

Ασκήσεις προς επίλυση

(1) Δίδεται το δένδρο με ρίζα:



Να βρεθούν:

i) Τα υποδένδρα της ρίζας του.

ii) Τα υποδένδρα των κόμβων v_2 και v_3.

iii) Τα επίπεδα των κόμβων v_2, v_{14}, v_{21} και v_{30}.

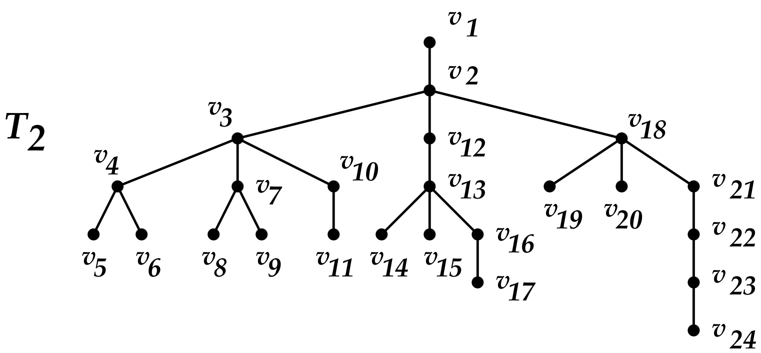
iv) Οι γονείς, τα παιδιά και τα αδέλφια των v_3 και v_{16}.

v) Όλοι οι πρόγονοι και όλοι οι απόγονοι του v_{19}.

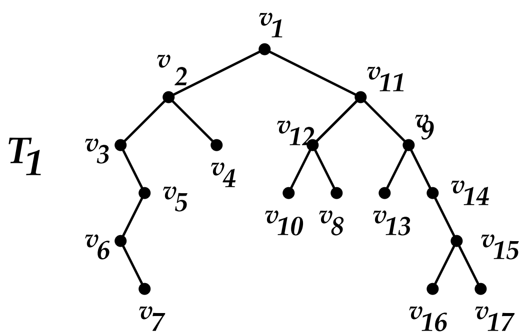
vi) Τα φύλλα του T.

vii) Το ύψος του T.

(2) Να γίνει διάσχιση του διατεταγμένου δένδρου T_2 και με τους 4 τρόπους.



(3) Να γίνει διάσχιση του δυαδικού δένδρου T_1 και του διατεταγμένου δένδρου T_2 και με τους 4 τρόπους.

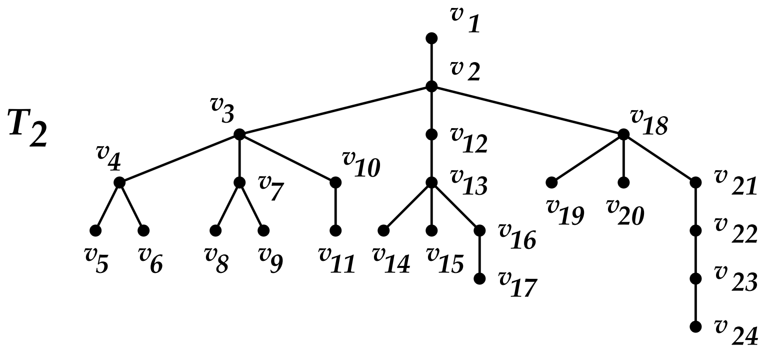


Προδιάταξη: $v_1, v_2, v_3, v_5, v_6, v_7, v_4, v_{11}, v_{12}, v_{10}, v_8, v_9, v_{13}, v_{14}, v_{15}, v_{16}, v_{17}$

Ενδοδιάταξη: $v_3, v_6, v_7, v_5, v_2, v_4, v_1, v_{10}, v_{12}, v_8, v_{11}, v_{13}, v_9, v_{14}, v_{16}, v_{15}, v_{17}$

Μεταδιάταξη: $v_7, v_6, v_5, v_3, v_4, v_2, v_{10}, v_8, v_{12}, v_{13}, v_{16}, v_{17}, v_{15}, v_{14}, v_9, v_{11}, v_1$

Διάταξη κατά επίπεδα: $v_1, v_2, v_{11}, v_3, v_4, v_{12}, v_9, v_5, v_{10}, v_8, v_{13}, v_{14}, v_6, v_{15}, v_7, v_{16}, v_{17}$



Προδιάταξη:

$v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}, v_{16}, v_{17}, v_{18}, v_{19}, v_{20}, v_{21}, v_{22}, v_{23}, v_{24}$

Ενδοδιάταξη:

$v_5, v_4, v_6, v_3, v_8, v_7, v_9, v_{11}, v_{10}, v_2, v_{14}, v_{13}, v_{15}, v_{17}, v_{16}, v_{12}, v_{19}, v_{18}, v_{20}, v_{24}, v_{23}, v_{22}, v_{21}, v_1$

Μεταδιάταξη:

$v_5, v_6, v_4, v_8, v_9, v_7, v_{11}, v_{10}, v_3, v_{14}, v_{15}, v_{17}, v_{16}, v_{13}, v_{12}, v_{19}, v_{20}, v_{24}, v_{23}, v_{22}, v_{21}, v_{18}, v_2, v_1$

Διάταξη κατά επίπεδα:

$v_1, v_2, v_3, v_{12}, v_{18}, v_4, v_7, v_{10}, v_{13}, v_{19}, v_{20}, v_{21}, v_5, v_6, v_8, v_9, v_{11}, v_{14}, v_{15}, v_{16}, v_{22}, v_{17}, v_{23}, v_{24}$

(4) Να δοθούν σε πολωνική γραφή οι παραστάσεις:

$$A = \frac{(\alpha+\beta)^2 - \gamma\delta}{\epsilon + \zeta\eta} - 3\left(\frac{\alpha}{\beta} - \gamma^3\right)$$

και

$$B = \frac{x - (y + \alpha\beta)}{(x+y)^2} - 2^x \frac{\alpha}{\beta + \gamma}$$

(5) Να βρεθεί η αλγεβρική παράσταση που αντιστοιχεί στην πολωνική γραφή:

$$+ / - * \alpha\beta\gamma - \delta * \epsilon\zeta * \uparrow + \eta\theta + \mu k \uparrow - \lambda 2 \nu$$

24. ΚΕΝΤΡΟ - ΚΕΝΤΡΟΕΙΔΕΣ ΔΕΝΔΡΟΥ

ΚΕΝΤΡΟ ΔΕΝΔΡΟΥ

Υπενθυμίζουμε ότι:

Η εκκεντρότητα ενός κόμβου v του συνεκτικού γραφήματος G ορίζεται από τη σχέση

$$e(v) = \max_{u \in V(G)} d(u, v).$$

Ένας κόμβος v ονομάζεται **κεντρικός** αν παρουσιάζει ελάχιστη εκκεντρότητα, δηλαδή αν

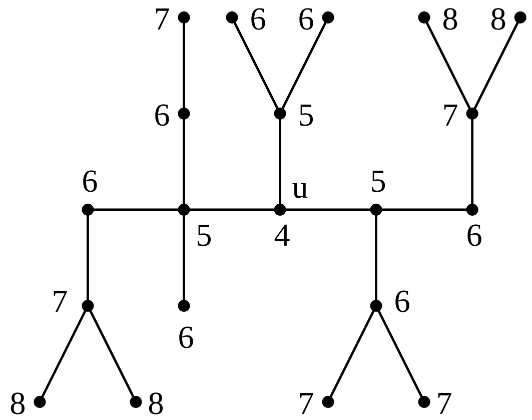
$$e(v) = \min_{u \in V(G)} e(u).$$

Μπορεί να υπάρχουν περισσότεροι του ενός κεντρικοί κόμβοι. Το σύνολο όλων των κεντρικών κόμβων ονομάζεται **κέντρο** του γραφήματος.

Ερώτηση Από πόσους κόμβους του δένδρου αποτελείται το κέντρο του;

Παραδείγματα

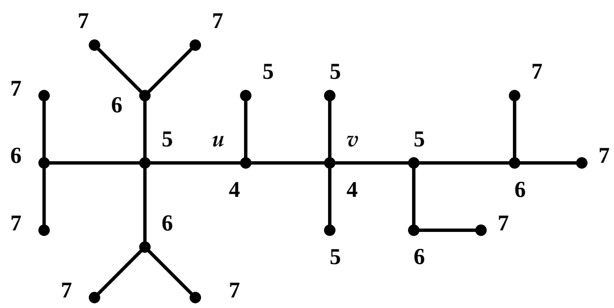
T_1:



Οι εκκεντρότητες των κορυφών του T_1

$\{u\}$: κέντρο (το κέντρο αποτελείται από ένα μόνο στοιχείο).

T_2:



Οι εκκεντρότητες των κορυφών του T_2

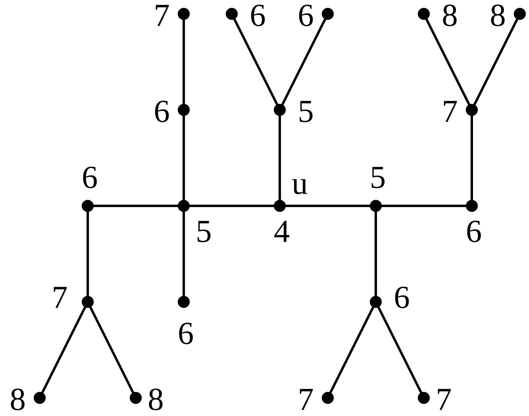
$\{u, v\}$: κέντρο (το κέντρο αποτελείται από δύο στοιχεία).

Πρόταση 43. *Το κέντρο ενός δένδρου T αποτελείται από έναν ή δύο (συνδεδεμένους μεταξύ τους) κόμβους.*

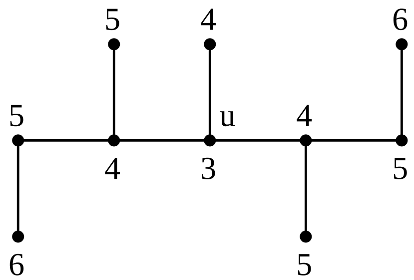
Η απόδειξη βασίζεται στην εξής παρατήρηση: Αν διαγράψουμε τα φύλλα ενός δένδρου T προκύπτει άλλο ένα δένδρο T' του οποίου οι κόμβοι έχουν εκκεντρότητα μειωμένη κατά 1 από αυτή που έχουν στο T και επομένως τα κέντρα των T, T' συμπίπτουν.

Παραδείγματα

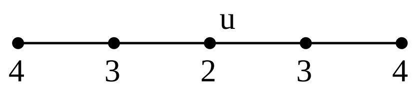
T_1:



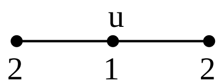
T_1':



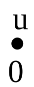
T_1'':



T_1''':

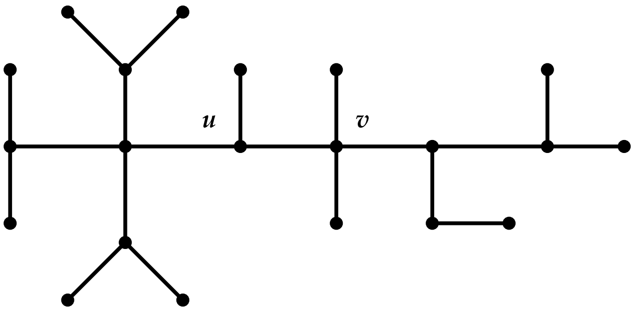


T_1'''':

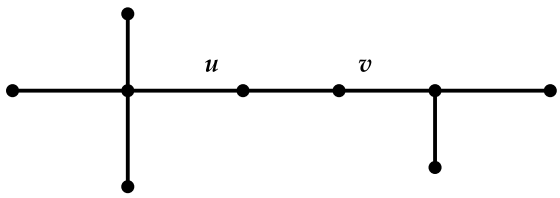


$\{u\}$: Κέντρο

T_2



T_2'



T_2''

T_2'''

$\{u, v\}$: Κέντρο

Απόδειξη της Πρότασης 43:
(Προφανής αν το T είναι το K_1 ή το K_2). Το δένδρο T' που προκύπτει από τη διαγραφή όλων των φύλλων του T έχει το ίδιο κέντρο με το T. Πράγματι, η διαγραφή αυτή μειώνει την εκκεντρότητα όλων των κόμβων που απομένουν στο T' κατά 1, (διότι για κάθε κόμβο το μέγιστο μονοπάτι που ξεκινά από αυτόν καταλήγει σε φύλλο). Άρα, όποιοι κόμβοι έχουν ελάχιστη εκκεντρότητα στο T, θα έχουν και στο T'. Άρα το T' έχει το ίδιο κέντρο με το T. Όμοια τώρα, διαγράφοντας όλα τα φύλλα του T', δημιουργούμε ένα δένδρο T'' με το ίδιο κέντρο, κ.ο.κ. μέχρι να παραμείνει ή το K_1 ή το K_2, που αποτελείται μόνο από 1 ή 2 κεντρικούς κόμβους, οι οποίοι αποτελούν και το κέντρο του αρχικού δένδρου T.

Πόρισμα 44. *Το κέντρο ενός δένδρου αποτελείται από δύο (συνδεδεμένους μεταξύ τους) κόμβους αν και μόνο αν η διάμετρός του είναι περιττή.*

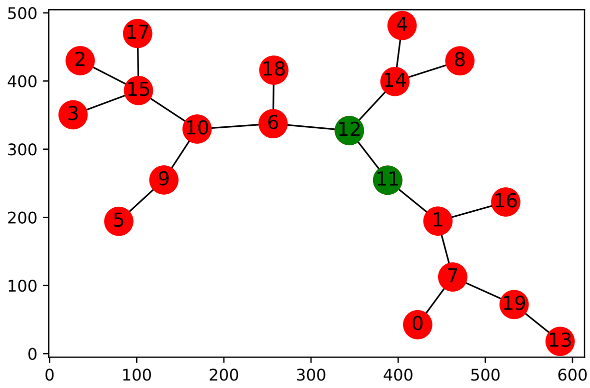
Ο επαναληπτικός αλγόριθμος που περιγράφεται στην απόδειξη της Πρότασης 43 μπορεί να υλοποιηθεί με την βιβλιοθήκη `networkx`.

```
import networkx as nx
import matplotlib.pyplot as plt

n = 20
T = nx.random_tree(n)
pos = nx.layout.kamada_kawai_layout(T)
nx.draw_networkx(T,pos)

Tcopy = nx.Graph(T) #copy tree T

```

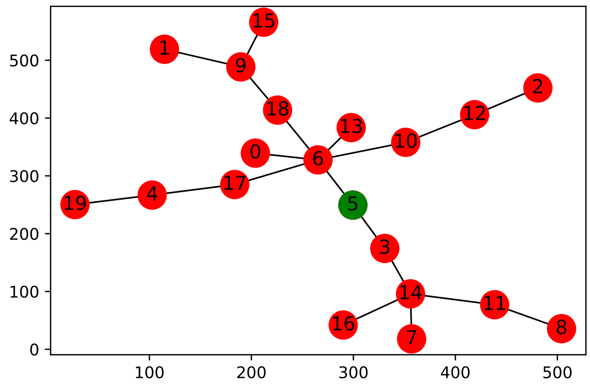
```
while(Tcopy.size() > 1): #until Tcopy has only one or zero edges
    #the set of remaining leaves in Tcopy
    leaves = []
    #for every vertex v in T
    for v in Tcopy:
        if(Tcopy.degree(v) == 1):     #v is a leaf
            leaves.append(v)
    Tcopy.remove_nodes_from(leaves)  #remove all leaves
#Now Tcopy consists of the center nodes
print("The center of T is:", Tcopy.nodes)
#Draw the center
nx.draw_networkx_nodes(Tcopy,pos,node_color='green',width=6.0)

plt.show()
```

Output:

```
The center of T is: [11, 12]
```



```
The center of T is: [5]
```



ΚΕΝΤΡΟΕΙΔΕΣ ΔΕΝΔΡΟΥ

Κλαδί ενός δένδρου T σε ένα κόμβο του v λέγεται κάθε μεγιστικό υποδένδρο του T που έχει το v για φύλλο. (Προφανώς ένα δένδρο έχει τόσα κλαδιά στο v όσος ο βαθμός του v).

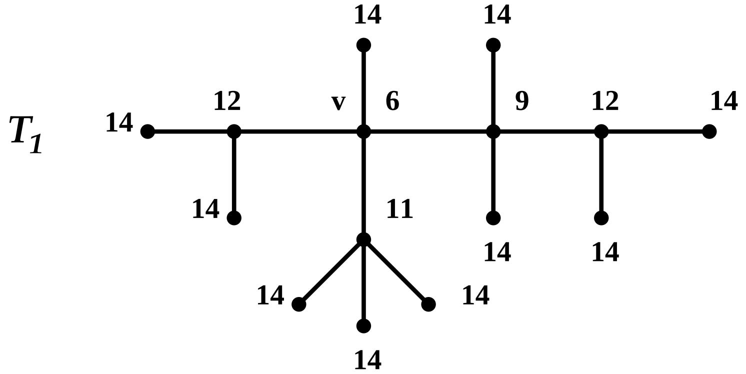
Βάρος ενός κλαδιού ονομάζεται ο αριθμός των δεσμών του.

Βάρος $w(v)$ ενός κόμβου v ονομάζεται το μέγιστο βάρος των κλαδιών στον κόμβο v.

Κεντροειδής κόμβος ενός δένδρου λέγεται κάθε κόμβος με ελάχιστο βάρος.

Κεντροειδές ενός δένδρου λέγεται το σύνολο των κεντροειδών κόμβων του.

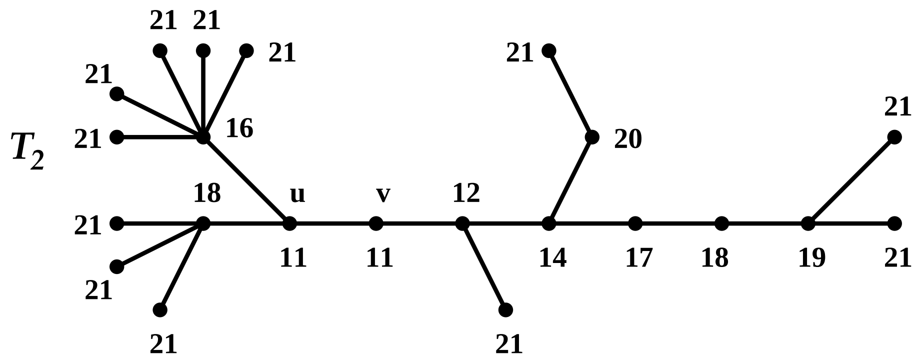
Παραδείγματα:



Τα βάρη των σημείων του T_1

v : κεντρειδές σημείο του T

$\{v\}$: κεντροειδές του T.



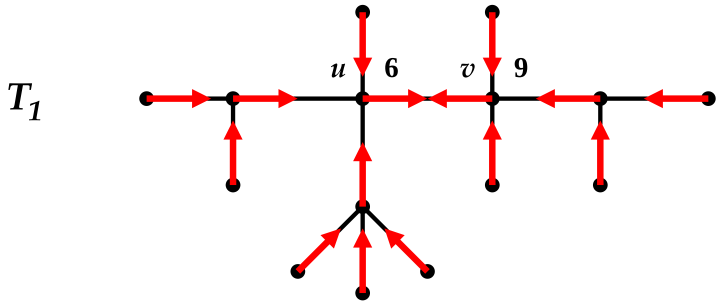
Τα βάρη των σημείων του T

u, v : κεντροειδή σημεία του T

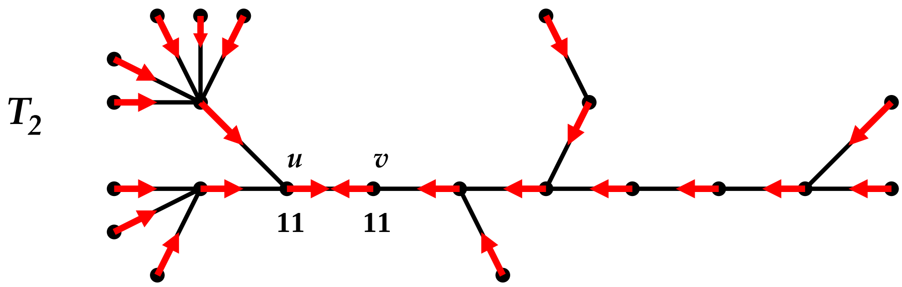
$\{u, v\}$: κεντροειδές του T.

Πρόταση 45. *Το κεντροειδές κάθε δένδρου αποτελείται από ένα ή δύο (συνδεδεμένους) κόμβους.*

Η απόδειξη που θα δοθεί στηρίζεται στην παρακάτω διαδικασία:
Για κάθε κόμβο u του δένδρου βρίσκουμε ένα κλαδί, του οποίου το βάρος ισούται με το βάρος του κόμβου. Στη συνέχεια βάζουμε ένα βελάκι πάνω στον πρώτο δεσμό του κλαδιού αυτού, με φορά προς το κλαδί. Τότε κάθε δεσμός θα έχει ένα ακριβώς βελάκι εκτός ενός δεσμού που θα έχει δύο βελάκια.
Αν τα άκρα του δεσμού που έχει δύο βελάκια έχουν το αυτό βάρος, το κεντροειδές του δένδρου θα αποτελείται από αυτά, διαφορετικά θα αποτελείται από το άκρο με το ελάχιστο βάρος.
Παραδείγματα

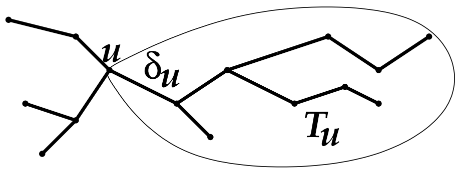


Όπως φαίνεται στο παραπάνω σχήμα ο δεσμός $\{u, v\}$ είναι ο μοναδικός δεσμός με δύο βελάκια. Επειδή $w(u) = 6 < 9 = w(v)$, το κεντροειδές του δένδρου θα είναι το $\{u\}$.



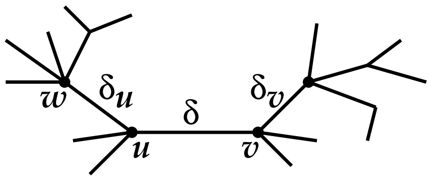
Όπως φαίνεται στο παραπάνω σχήμα ο δεσμός $\{u, v\}$ είναι ο μοναδικός δεσμός με δύο βελάκια. Επειδή $w(u) = 11 = w(v)$, το κεντροειδές του δένδρου είναι το $\{u, v\}$.

Απόδειξη της Πρότασης 45: Σε κάθε κόμβο του δένδρου αντιστοιχούμε ένα δεσμό δ_u ως εξής: Επιλέγουμε ένα κλαδί T_u του κόμβου u που το βάρος του δίνει το βάρος του κόμβου και ορίζουμε δ_u το δεσμό του κλαδιού αυτού που περιέχει το u.



$$w(u) = \text{αριθμός δεσμών } T_u = 9.$$

Για κάθε δεσμό $\delta = \{u, v\}$ ισχύει $\delta = \delta_u$ ή $\delta = \delta_v$ διότι διαφορετικά το δ δε θα ανήκει στα κλαδιά T_u και T_v,



οπότε

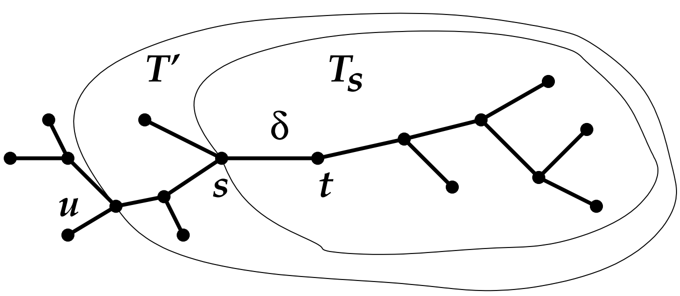
$$\begin{aligned} w(u) &= \text{βάρος } T_u \\ &\geq \text{βάρος του κλαδιού του } u \text{ με πρώτο δεσμό το } \delta \\ &> w(v) \end{aligned}$$

και ομοίως

$$\begin{aligned} w(v) &= \text{βάρος } T_v \\ &\geq \text{βάρος του κλαδιού του } v \text{ με πρώτο δεσμό το } \delta \\ &> w(u) \end{aligned}$$

το οποίο είναι άτοπο.

Άρα, κάθε δεσμός δ αντιστοιχεί σε ένα ή δύο κόμβους u (άκρα του δ) με $\delta = \delta_u$. Επειδή ο αριθμός των κόμβων του δένδρου είναι μεγαλύτερος κατά ένα από τον αριθμό των δεσμών ($|X| = |E| + 1$) θα υπάρχει ακριβώς ένας δεσμός $\delta = \{s, t\}$ με $\delta = \delta_s = \delta_t$. Θα δειχθεί ότι για κάθε $u \in X$ με $u \neq s, t$ ισχύει: Αν u ανήκει στην ίδια συνιστώσα με το s (αντίστοιχα t) στο $T - \{s, t\}$ τότε $w(u) > w(s)$ (αντίστοιχα $w(u) > w(t)$). Πράγματι αν θεωρήσουμε χωρίς βλάβη της γενικότητας ότι το u ανήκει στην ίδια συνιστώσα με το s στο $T - \{s, t\}$ τότε το T_s θα είναι υποδένδρο του T' (βλέπε επόμενο σχήμα).



Άρα

$$\begin{aligned} w(s) &= \text{αριθμός κόμβων του δένδρου } T_s \\ &< \text{αριθμός κόμβων του δένδρου } T' \\ &\leq w(u) \end{aligned}$$

Άρα $\min\{w(s), w(t)\} = \min_{u \in X} w(u)$ και επομένως το κεντροειδές K' είναι

$$K' = \begin{cases} \{s\}, & \text{αν } w(s) < w(t) \\ \{t\}, & \text{αν } w(t) < w(s) \\ \{s,t\}, & \text{αν } w(s) = w(t). \end{cases}$$

□

Χρησιμοποιώντας την βιβλιοθήκη **networkx** μπορούμε να υλοποιήσουμε μια μέθοδο υπολογισμού του κεντροειδούς ενός δένδρου. Ο υπολογισμός του κεντροειδούς προκύπτει από τον υπολογισμό των βαρών κάθε κόμβου του δένδρου, βρίσκοντας το βάρος όλων των κλαδιών κάθε κόμβου.

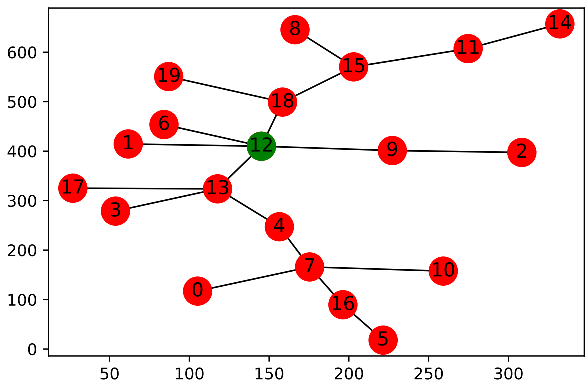
```
import networkx as nx
import matplotlib.pyplot as plt

n = 20
T = nx.random_tree(n)
pos = nx.layout.kamada_kawai_layout(T)
nx.draw_networkx(T,pos)

#dictionary of weights
node_weights = {}
#minimum weight of nodes
min_node_weight = n-1

for v in T:
    Tcopy = nx.Graph(T) #copy tree T
    Tcopy.remove_node(v)
    #Tcopy consists of the branches of nodes v
    #(except the edge that connects it with v)
    #Each branch is a connected component of Tcopy
    Tcc = nx.connected_components(Tcopy)
    #compute the weight of v
    weight = 0
    for cc in Tcc:
        #len(cc) = the weight of branch cc
        if(len(cc) > weight):
            weight = len(cc)
    node_weights[v] = weight
    #compute the min weight of nodes
    if(min_node_weight > weight):
        min_node_weight = weight

centroid = []
for v in T:
    if node_weights[v] == min_node_weight:
        centroid.append(v)

```

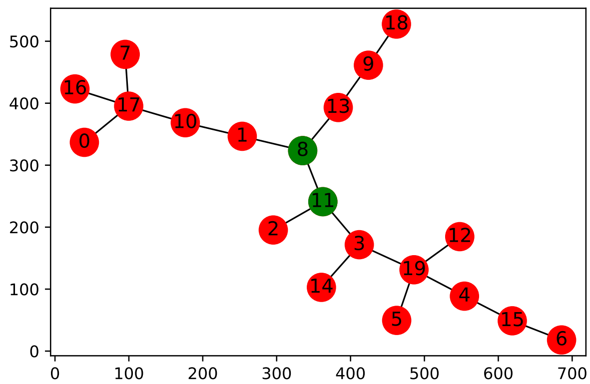
```
print("The centroid of T is:",centroid)
print("The centroid nodes have weights:",min_node_weight)

#Draw the center
Tcentroid = T.subgraph(centroid)
nx.draw_networkx_nodes(Tcentroid,pos,node_color='green',width=6.0)

plt.show()
print("The node weights are:", node_weights)
```

Output:

```
The centroid of T is: [12]
The centroid nodes have weights: 9
```

```
The centroid of T is: [8, 11]
The centroid nodes have weights: 10
```





```
The node weights are: {0: 19, 1: 19, 2:
    19, 3: 19, 4: 14, 5: 19, 6: 19, 7: 15,
     8: 19, 9: 18, 10: 19, 11: 18, 12: 9,
    13: 11, 14: 19, 15: 16, 16: 18, 17:
    19, 18: 14, 19: 19}
```

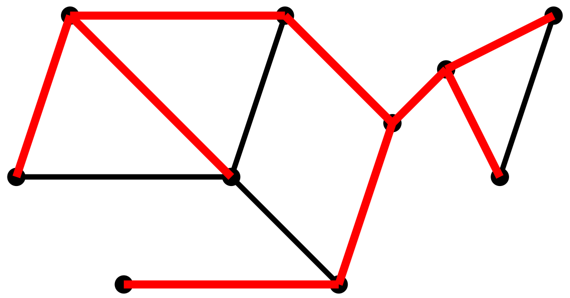
```
The node weights are: {0: 19, 1: 14, 2:
    19, 3: 12, 4: 17, 5: 19, 6: 19, 7: 19,
     8: 10, 9: 18, 10: 15, 11: 10, 12: 19,
     13: 17, 14: 19, 15: 18, 16: 19, 17:
    16, 18: 19, 19: 14}
```

25. ΔΕΝΔΡΟ ΖΕΥΞΗΣ ΓΡΑΦΗΜΑΤΟΣ

Υπενθυμίζουμε ότι ένα γράφημα $G_1 = (X, E_1)$ λέγεται γενετικό υπογράφημα ενός γραφήματος $G = (X, E)$, αν $E_1 \subseteq E$. Αν το G_1 είναι δένδρο, τότε έχουμε ένα **δένδρο ζεύξης** (spanning tree), **ή γενετικό (ή γεννητικό) δένδρο**, του G.

Παράδειγμα : Οι έντονες γραμμές δίνουν στο παρακάτω γράφημα G ένα γενετικό δένδρο. (Φυσικά μπορούμε να βρούμε κι άλλα γενετικά δένδρα για το ίδιο G).

G



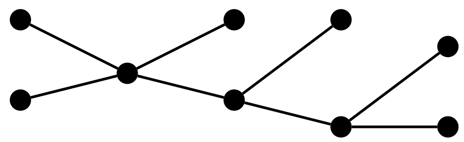
Ένα γενετικό δένδρο του G.

Πρόταση 46. *Ένα γράφημα έχει (τουλάχιστον ένα) δένδρο ζεύξης αν και μόνο αν είναι συνεκτικό.*

Παρατήρηση: Κάθε δένδρο ζεύξης ενός συνεκτικού γραφήματος με n κορυφές αποτελείται από $n-1$ δεσμούς του γραφήματος οι οποίοι δεν σχηματίζουν κύκλο.

Άσκηση 10. *Να βρεθεί το πλήθος των δένδρων ζεύξης των παρακάτω γραφημάτων:*

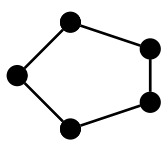
T



i)

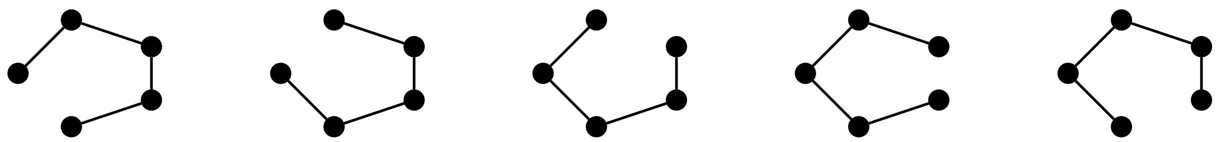
Λύση. Το T έχει 1 δένδρο ζεύξης, τον εαυτό του. □

C_5



ii)

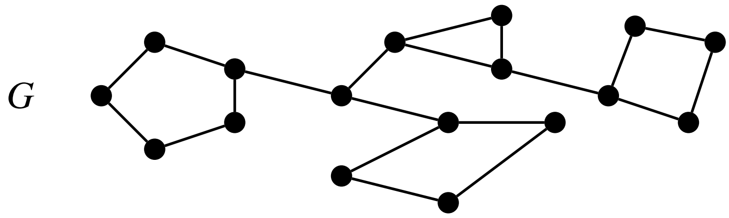
Λύση. Το C_5 έχει 5 δένδρα ζεύξης:



□

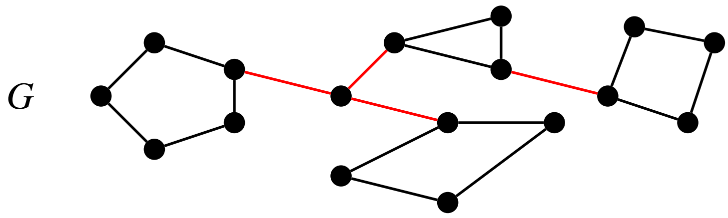
iii) C_n

Λύση. Το C_n έχει n δένδρα ζεύξης. Το καθένα προκύπτει σβήνοντας ακριβώς ένα δεσμό του C_n. □

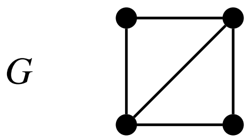


iv)

Λύση. Οι γέφυρες του γραφήματος (δεσμοί που είναι σημειωμένοι με κόκκινο) πρέπει να ανήκουν σε όλα τα δένδρα ζεύξης.

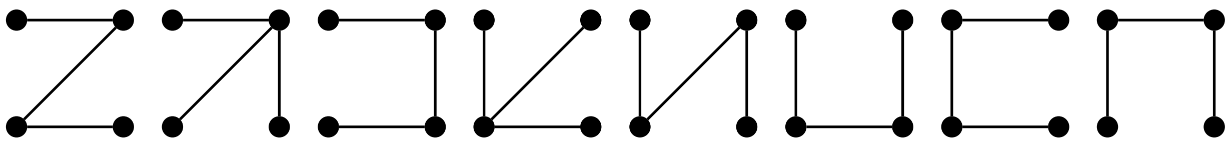


Σε κάθε ένα από τους 4 κύκλους που εμφανίζονται στο G μπορούμε να επιλέξουμε ανεξάρτητα ένα δένδρο ζεύξης τους, οι δεσμοί του οποίου περιέχονται στο δένδρο ζεύξης του G. Επομένως, το G έχει $5 \cdot 4 \cdot 3 \cdot 4 = 240$ δένδρα ζεύξης. □



v)

Λύση. Το G έχει 4 κορυφές άρα κάθε δένδρο ζεύξης του πρέπει να περιέχει 3 δεσμούς. Υπάρχουν 5 δεσμοί στο G από τους οποίους πρέπει να επιλέξουμε 3. Άρα, ο αριθμός των δένδρων ζεύξης του G είναι το πολύ $\binom{5}{3} = 10$. Όμως, κάποιες τριάδες δεσμών του G δεν αντιστοιχούν σε δένδρο. Οπότε τελικά έχουμε τα παρακάτω 8 δένδρα ζεύξης του G.

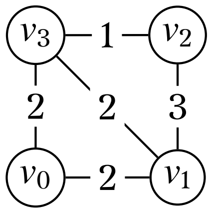


□

Παρατήρηση *Για το πλήθος των δένδρων ζεύξης ενός γραφήματος G υπάρχει τύπος που απαιτεί τον υπολογισμό μιας ορίζουσας (matrix-tree theorem). Η πολυπλοκότητα του τύπου είναι πολυωνυμική ως προς τον αριθμό των κορυφών του γραφήματος.*

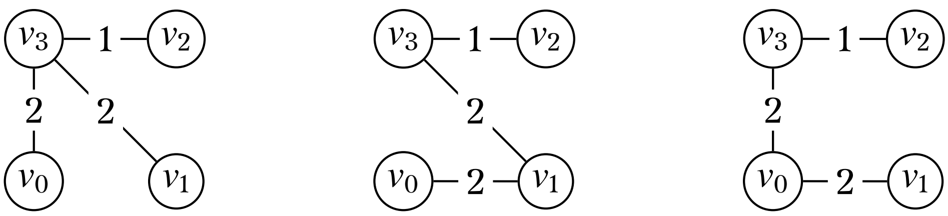
26. ΕΛΑΧΙΣΤΑ ΔΕΝΔΡΑ ΖΕΥΞΗΣ

Πρόβλημα. Δίδεται ένα συνεκτικό γράφημα οι δεσμοί του οποίου έχουν βάρη.



Να βρεθεί ένα υποσύνολο των δεσμών του με την ιδιότητα ότι υπάρχει μονοπάτι που συνδέει οποιεσδήποτε κορυφές του και το συνολικό άθροισμα των βαρών του είναι ελάχιστο.

Προφανώς στο ζητούμενο γράφημα περιέχονται όλες οι κορυφές του αρχικού γραφήματος, το γράφημα που προκύπτει είναι συνεκτικό, αφού υπάρχει μονοπάτι που συνδέει όλες τις κορυφές του και επιπλέον δεν υπάρχουν κύκλοι, αφού αν διαγράψουμε ένα δεσμό ενός κύκλου διατηρείται η συνεκτικότητα του γραφήματος.
Επομένως το ζητούμενο είναι η εύρεση ενός δένδρου με το ελάχιστο συνολικό κόστος βαρών. Το δένδρο αυτό ονομάζεται ελάχιστο δένδρο ζεύξης, ή ελάχιστο γεννητικό δένδρο, ή ελάχιστο επικαλύπτον δένδρο, (ή minimum spanning tree (MST)).
Το ελάχιστο δένδρο ζεύξης δεν είναι απαραίτητα μοναδικό, για το προηγούμενο γράφημα υπάρχουν 3 ελάχιστα δένδρα ζεύξης με κόστος 5 το καθένα.



Σε κάθε δένδρο με $|V|$ κορυφές και $|E|$ δεσμούς ισχύει ότι

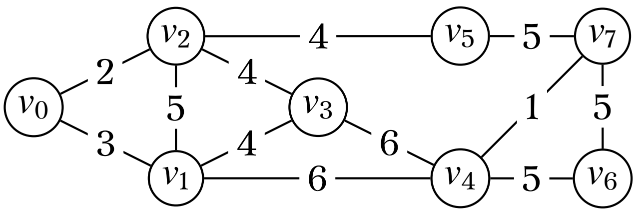
$$|V| = |E| + 1$$

Επομένως, αν το γράφημα έχει n κορυφές τότε το ελάχιστο δένδρο ζεύξης θα έχει $n-1$ δεσμούς. Απομένει να βρεθεί ποιοι δεσμοί πρέπει να επιλεγούν.
Στη συνέχεια παρουσιάζεται ένας αλγόριθμος που επιλύει αποτελεσματικά αυτό το πρόβλημα.

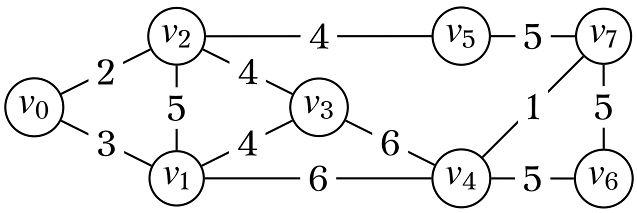
26.1. Ο αλγόριθμος του Kruskal.

> **Αλγόριθμος του Kruskal**
> Είσοδος: Ένα γράφημα δεσμών με $|V|$ κορυφές και βάρη.
> Έξοδος: Ένα ελάχιστο δένδρο ζεύξης (MST).
> (1) Τοποθετούμε όλους τους δεσμούς σε μια ουρά προτεραιότητας Q, με κλειδί το βάρος κάθε δεσμού.
> (2) Όσο δεν έχουν επιλεγεί $|V| - 1$ δεσμοί εκτέλεσε τα επόμενα βήματα:
> - Αφαίρεσε από την Q τον δεσμό με το μικρότερο βάρος.
> - Αν η επιλογή του δεν δημιουργεί κύκλο προσθεσέ τον στο ελάχιστο δένδρο ζεύξης.

Παράδειγμα
Να βρεθεί, με τον αλγόριθμο του Kruskal, ένα ελάχιστο δένδρο ζεύξης για το επόμενο γράφημα, το οποίο έχει 8 κορυφές και 12 δεσμούς.



(1) Αρχικά τοποθετούμε τους 12 δεσμούς στην ουρά προτεραιότητας Q, και το δένδρο MST είναι κενό.

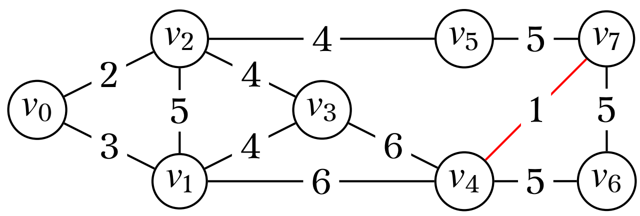


$$Q = [\{0,2\}[2], \{0,1\}[3], \{1,2\}[5], \{2,3\}[4], \{1,3\}[4], \{1,4\}[6], \{3,6\}[6], \{2,5\}[4], \{5,7\}[5], \{6,7\}[5], \{4,7\}[1], \{4,6\}[5]]$$

$$MST = \varnothing$$

(2) Επειδή έχουμε επιλέξει μόνο 0 δεσμούς < 7, αφαιρούμε από την ουρά προτεραιότητας τον δεσμό με το ελάχιστο κλειδί, δηλαδή τον δεσμό $\{4,7\}$ με βάρος 1.

Η προσθήκη του δεσμού $\{4,7\}$ στο δένδρο MST δεν δημιουργεί κύκλο, άρα ανήκει στο MST.

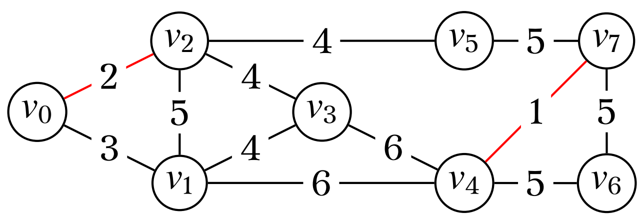


$$Q = [\{0,2\}[2], \{0,1\}[3], \{1,2\}[5], \{2,3\}[4], \{1,3\}[4], \{1,4\}[6], \{3,6\}[6],$$
$$\{2,5\}[4], \{5,7\}[5], \{6,7\}[5], \{4,6\}[5]$$

$$MST = [\{4,7\}[1]]$$

(3) Επειδή έχουμε επιλέξει μόνο 1 δεσμό < 7, αφαιρούμε από την ουρά προτεραιότητας τον δεσμό με το ελάχιστο κλειδί, δηλαδή τον δεσμό $\{0,2\}$ με βάρος 2.

Η προσθήκη του δεσμού $\{0,2\}$ στο δένδρο MST δεν δημιουργεί κύκλο, άρα ανήκει στο MST.

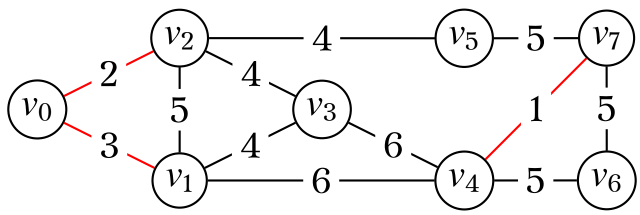


$$Q = [\{0,1\}[3], \{1,2\}[5], \{2,3\}[4], \{1,3\}[4], \{1,4\}[6], \{3,6\}[6], \{2,5\}[4],$$
$$\{5,7\}[5], \{6,7\}[5], \{4,6\}[5]$$

$$MST = [\{0,2\}[2], \{4,7\}[1]]$$

(4) Επειδή έχουμε επιλέξει μόνο 2 δεσμούς < 7, αφαιρούμε από την ουρά προτεραιότητας τον δεσμό με το ελάχιστο κλειδί, δηλαδή τον δεσμό $\{0,1\}$ με βάρος 3.

Η προσθήκη του δεσμού $\{0,1\}$ στο δένδρο MST δεν δημιουργεί κύκλο, άρα ανήκει στο MST.

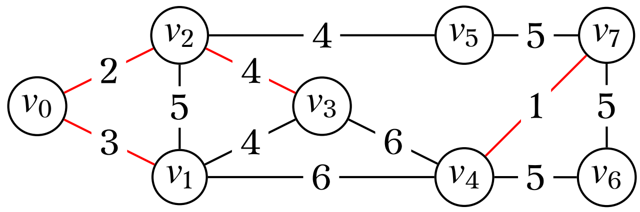


$$Q = [\{1,2\}[5], \{2,3\}[4], \{1,3\}[4], \{1,4\}[6], \{3,6\}[6], \{2,5\}[4], \{5,7\}[5], \{6,7\}[5], \{4,6\}[5]]$$

$$MST = [\{0,1\}[3], \{0,2\}[2], \{4,7\}[1]]$$

(5) Επειδή έχουμε επιλέξει μόνο 3 δεσμούς < 7, αφαιρούμε από την ουρά προτεραιότητας τον δεσμό με το ελάχιστο κλειδί, υπάρχουν τρεις δεσμοί με ελάχιστο κλειδί ίσο με 4, οι δεσμοί $\{2,3\}$, $\{1,3\}$ και $\{2,5\}$. Διαλέγουμε στην τύχη τον δεσμό $\{2,3\}$.

Η προσθήκη του δεσμού $\{2,3\}$ στο δένδρο MST δεν δημιουργεί κύκλο, άρα ανήκει στο MST.

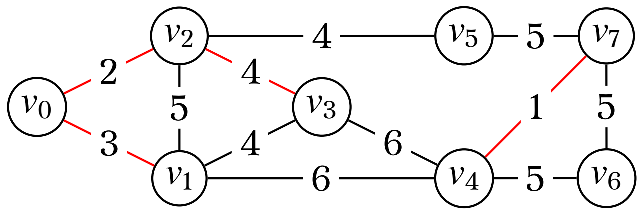


$$Q = [\{1,2\}[5], \{1,3\}[4], \{1,4\}[6], \{3,6\}[6], \{2,5\}[4], \{5,7\}[5], \{6,7\}[5], \{4,6\}[5]]$$

$$MST = [\{4,7\}[1], \{0,2\}[2], \{0,1\}[3], \{2,3\}[4]]$$

(6) Επειδή έχουμε επιλέξει μόνο 4 δεσμούς < 7, αφαιρούμε από την ουρά προτεραιότητας τον δεσμό με το ελάχιστο κλειδί, υπάρχουν δύο δεσμοί με ελάχιστο κλειδί ίσο με 4, οι δεσμοί $\{1,3\}$ και $\{2,5\}$. Διαλέγουμε στην τύχη τον δεσμό $\{1,3\}$.

Η προσθήκη του δεσμού $\{1,3\}$ στο δένδρο MST **δημιουργεί κύκλο**, άρα δεν ανήκει στο MST.

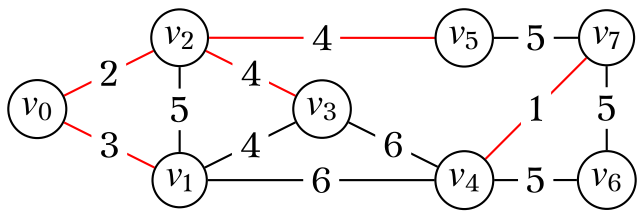


$$Q = [\{1,2\}[5], \{1,4\}[6], \{3,6\}[6], \{2,5\}[4], \{5,7\}[5], \{6,7\}[5], \{4,6\}[5]]$$

$$MST = [\{4,7\}[1], \{0,2\}[2], \{0,1\}[3], \{2,3\}[4]]$$

(7) Επειδή έχουμε επιλέξει μόνο 4 δεσμούς < 7, αφαιρούμε από την ουρά προτεραιότητας τον δεσμό με το ελάχιστο κλειδί, δηλαδή τον δεσμό $\{2,5\}$ με βάρος 4.

Η προσθήκη του δεσμού $\{2,5\}$ στο δένδρο MST δεν δημιουργεί κύκλο, άρα ανήκει στο MST.

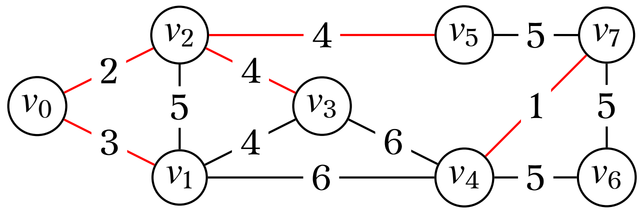


$$Q = [\{1,2\}[5], \{1,4\}[6], \{3,6\}[6], \{5,7\}[5], \{6,7\}[5], \{4,6\}[5]]$$

$$MST = [\{4,7\}[1], \{0,2\}[2], \{0,1\}[3], \{2,3\}[4], \{2,5\}[4]]$$

(8) Επειδή έχουμε επιλέξει μόνο 5 δεσμούς < 7, αφαιρούμε από την ουρά προτεραιότητας τον δεσμό με το ελάχιστο κλειδί, υπάρχουν τέσσερεις δεσμοί με ελάχιστο κλειδί ίσο με 5, οι δεσμοί $\{1,2\}$, $\{5,7\}$, $\{6,7\}$ και $\{4,6\}$. Διαλέγουμε στην τύχη τον δεσμό $\{1,2\}$.

Η προσθήκη του δεσμού $\{1,2\}$ στο δένδρο MST **δημιουργεί κύκλο**, άρα δεν ανήκει στο MST.

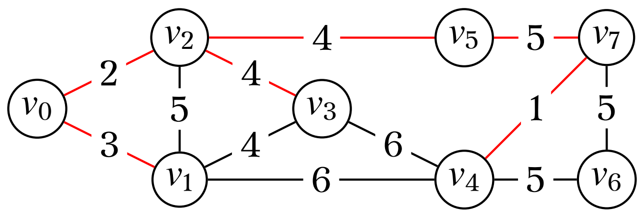


$$Q = [\{1,4\}[6], \{3,6\}[6], \{5,7\}[5], \{6,7\}[5], \{4,6\}[5]]$$

$$MST = [\{4,7\}[1], \{0,2\}[2], \{0,1\}[3], \{2,3\}[4], \{2,5\}[4]]$$

(9) Επειδή έχουμε επιλέξει μόνο 5 δεσμούς < 7, αφαιρούμε από την ουρά προτεραιότητας τον δεσμό με το ελάχιστο κλειδί, υπάρχουν τρεις δεσμοί με ελάχιστο κλειδί ίσο με 5, οι δεσμοί $\{5,7\}$, $\{6,7\}$ και $\{4,6\}$. Διαλέγουμε στην τύχη τον δεσμό $\{5,7\}$.

Η προσθήκη του δεσμού $\{5,7\}$ στο δένδρο MST δεν δημιουργεί κύκλο, άρα ανήκει στο MST.

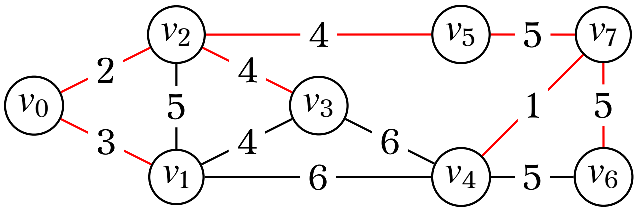


$$Q = [\{1,4\}[6], \{3,6\}[6], \{6,7\}[5], \{4,6\}[5]]$$

$$MST = [\{4,7\}[1], \{0,2\}[2], \{0,1\}[3], \{2,3\}[4], \{2,5\}[4], \{5,7\}[5]]$$

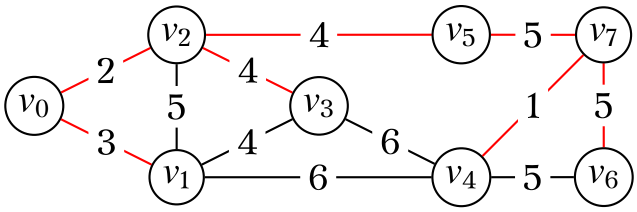
(10) Επειδή έχουμε επιλέξει μόνο 6 δεσμούς < 7, αφαιρούμε από την ουρά προτεραιότητας τον δεσμό με το ελάχιστο κλειδί, υπάρχουν δύο δεσμοί με ελάχιστο κλειδί ίσο με 5, οι δεσμοί $\{6,7\}$ και $\{4,6\}$. Διαλέγουμε στην τύχη τον δεσμό $\{6,7\}$.

Η προσθήκη του δεσμού $\{6,7\}$ στο δένδρο MST δεν δημιουργεί κύκλο, άρα ανήκει στο MST.



$$Q = [\{1,4\}[6], \{3,6\}[6], \{4,6\}[5]]$$
$$MST = [\{4,7\}[1], \{0,2\}[2], \{0,1\}[3], \{2,3\}[4], \{2,5\}[4], \{5,7\}[5], \{6,7\}[5]]$$

(11) Επειδή έχουμε επιλέξει 7 δεσμούς, το δένδρο MST είναι το ελάχιστο δένδρο ζεύξης, με συνολικό κόστος $1 + 2 + 3 + 4 + 4 + 5 + 5 = 24$.



$$Q = [\{1,4\}[6], \{3,6\}[6], \{4,6\}[5]]$$
$$MST = [\{4,7\}[1], \{0,2\}[2], \{0,1\}[3], \{2,3\}[4], \{2,5\}[4], \{5,7\}[5], \{6,7\}[5]]$$

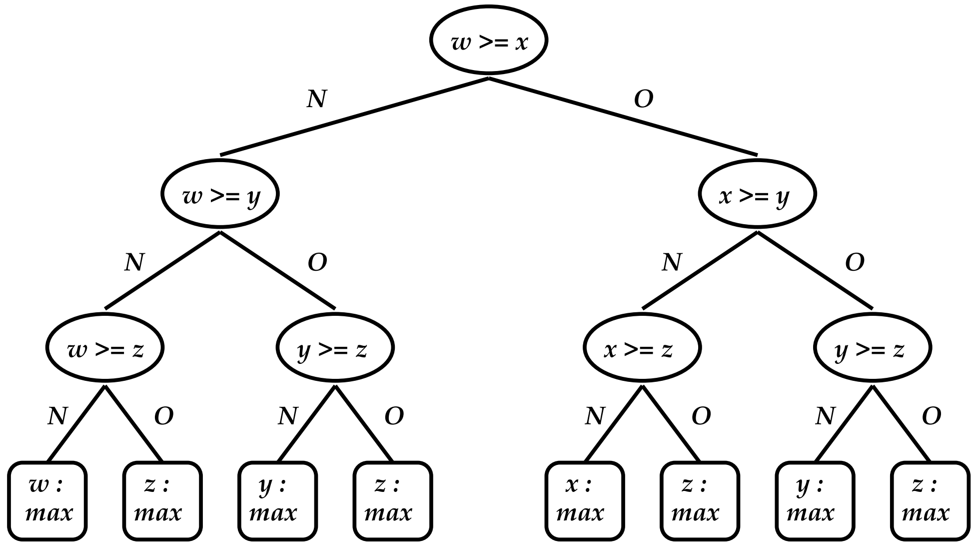
Παρατηρήσεις.

- Στον αλγόριθμο του Kruskal το ελάχιστο δένδρο ζεύξης δημιουργείται συνδέοντας μικρότερα ελάχιστα δένδρα ζεύξης (δάση από δένδρα ζεύξης).
- Στον αλγόριθμο του Kruskal το βαρύτερο υπολογιστικό κομμάτι είναι ο έλεγχος της δημιουργίας κύκλου. Υπάρχουν αποδοτικές δομές δεδομένων για τον έλεγχο αυτό.

27. ΔΕΝΔΡΑ ΑΠΟΦΑΣΕΩΝ

Δένδρο απόφασης (decision tree) λέγεται ένα k-δένδρο του οποίου κάθε εσωτερική κορυφή παριστάνει μια ερώτηση για την οποία πρέπει να αποφασίσουμε. Οι δυνατές απαντήσεις (αποφάσεις) παριστάνονται από τους δεσμούς που συνδέουν τον κόμβο με τους κόμβους του επόμενου επίπεδου. Τα τελικά αποτελέσματα της διαδικασίας παριστάνονται από τα φύλλα του δένδρου.

Πολύ συχνά, οι δυνατές απαντήσεις κάθε φορά είναι : "Ναι" (Ν) ή "Όχι " (Ο), οπότε το δένδρο απόφασης είναι ένα δυαδικό δένδρο, όπως το παρακάτω δένδρο που βρίσκει το μέγιστο μεταξύ 4 στοιχείων x, y, z, w χρησιμοποιώντας 3 συγκρίσεις.

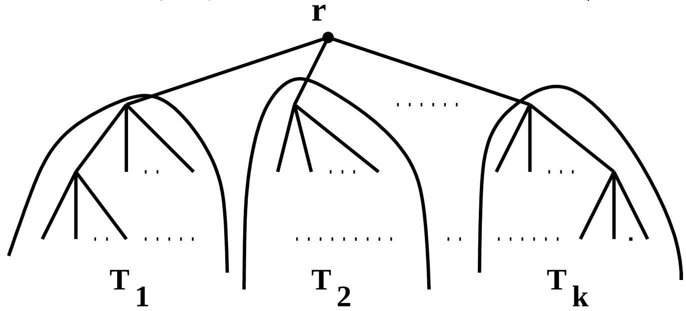


Ένα πολύ χρήσιμο θεώρημα που μας δίνει κάτω φράγματα για το πλήθος των ερωτήσεων που απαιτούνται σε κάποιο πρόβλημα απόφασης που μοντελοποιείται από ένα k-δένδρο είναι το επόμενο.

Πρόταση 47. *Το ύψος h ενός k-δένδρου με l φύλλα είναι τουλάχιστον* $\log_k l$.

Απόδειξη. Θέλουμε να δείξουμε ότι $h \geq \log_k l$, ή ισοδύναμα ότι $k^h \geq l$, δηλαδή θέλουμε να δείξουμε ότι ένα k-δένδρο ύψους h έχει το πολύ k^h φύλλα. Χρησιμοποιούμε επαγωγή ως προς h: Για $h = 0$, το δένδρο είναι τετριμμένο και ο μοναδικός του κόμβος είναι και φύλλο, άρα έχει πράγματι k^0 φύλλα.

Έστω ότι ισχύει για $h \leq n$, και έστω T ενα k-δένδρο ύψους $n + 1$. Στο γράφημα $T - r$ (όπου r η ρίζα του T) έχουμε k το πολύ υποδένδρα, τα $T_1, T_2, \ldots, T_k$.



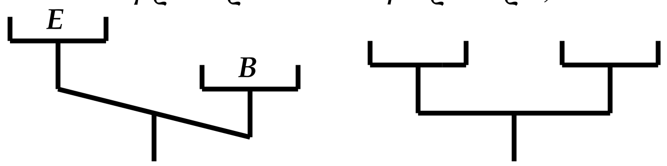
Αλλά τα $T_1, T_2, \ldots, T_k$ έχουν ύψος το πολύ n. Άρα, από την υπόθεση της επαγωγής, έχουν συνολικά k^n το πολύ φύλλα το καθένα. Συνολικά λοιπόν, τα υποδένδρα $T_1, T_2, \ldots, T_k$ έχουν το πολύ $k \cdot k^n = k^{n+1}$ φύλλα, τα οποία είναι τα φύλλα και του αρχικού δένδρου T. $\square$

Παρατήρηση: Από την προηγούμενη πρόταση προκύπτει ότι ένα k-δένδρο με ύψος h περιέχει το πολύ k^h φύλλα.

Στο επόμενο παράδειγμα δίνεται μια εφαρμογή που χρησιμοποιεί ένα 3-δένδρο απόφασης:

Άσκηση 11 (Το πρόβλημα των κίβδηλων νομισμάτων). *Ένα κανονικό νόμισμα έχει αριθμό* 0*. Υπάρχουν n άλλα νομίσματα ίδια ακριβώς στην εμφάνιση με το* 0*, που έχουν όμως αριθμούς* $1, 2, \ldots, n$*. Υποψιαζόμαστε ότι ένα νόμισμα μπορεί να είναι "κίβδηλο" (είτε λίγο ελαφρύτερο, είτε λίγο βαρύτερο).*

i) Να δειχθεί ότι χρειάζονται τουλάχιστον $\log_3(2n+1)$ *ζυγίσματα σε μια ζυγαριά η οποία δείχνει το ελαφρύτερο και το βαρύτερο, ή δύο ίσου βάρους*



για να αποφασίσουμε αν υπάρχει κίβδηλο νόμισμα, ποιο είναι αυτό και αν είναι βαρύ ή ελαφρύ.

ii) Να περιγραφεί μια διαδικασία που να χρησιμοποιεί ακριβώς αυτό τον αριθμό ζυγισμάτων, όταν $n = 3$ *και όταν* $n = 4$*.*

Λύση. i) Το δένδρο απόφασης εδώ είναι ένα 3-δένδρο, αφού κάθε φορά που ζυγίζουμε υπάρχουν τρία πιθανά αποτελέσματα :

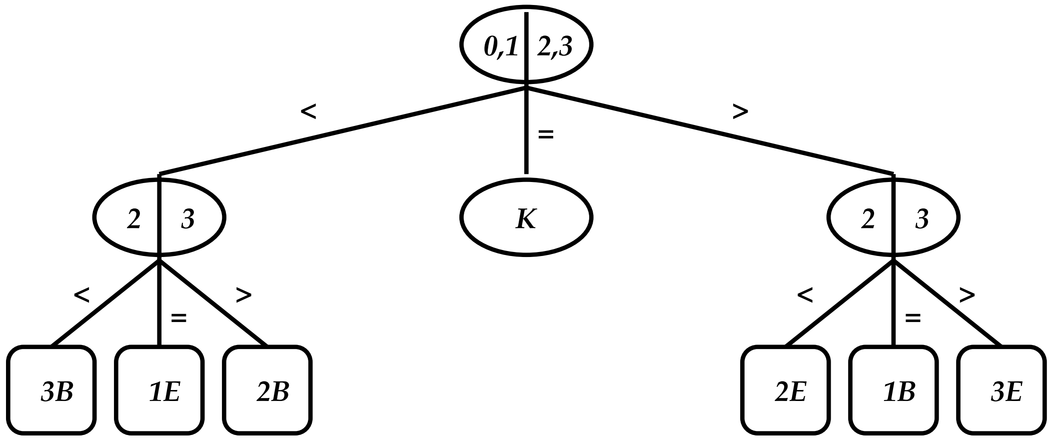
$<$: το αριστερό είναι ελαφρύτερο,
$=$: τα δύο μέρη έχουν το ίδιο βάρος,
$>$: το αριστερό είναι βαρύτερο.

Υπάρχουν $2n+1$ πιθανές τελικές απαντήσεις (φύλλα) στο δένδρο απόφασης, για το παραπάνω πρόβλημα

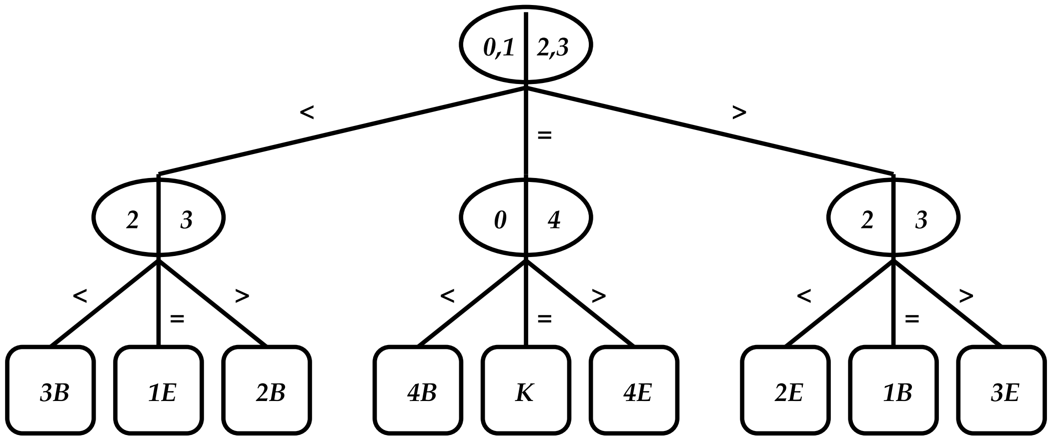
K,	$1B$,	$1E$,	$2B$,	$2E$,	$\ldots$,	nB,	nE
καλά όλα	το 1 είναι βαρύ	το 1 είναι ελαφρύ	το 2 είναι βαρύ	το 2 είναι ελαφρύ		το n είναι βαρύ	το n είναι ελαφρύ

Άρα το ύψος του δένδρου είναι τουλάχιστον $\log_3(2n+1)$, οπότε το πλήθος των δεσμών του δένδρου (δηλαδή των ζυγισμάτων) από τη ρίζα μέχρι κάποιο (τουλάχιστον ένα) φύλλο (δηλαδή σε μία τουλάχιστον περίπτωση) είναι τουλάχιστον $\log_3(2n+1)$.

ii) Για $n = 3$, έχουμε $\log_3(2 \cdot 3 + 1) = \log_3 7 \approx 1.771$, οπότε τα ζυγίσματα θα είναι τουλάχιστον 2, μια δε λύση με ακριβώς δύο ζυγίσματα φαίνεται στο παρακάτω 3-δένδρο αποφάσεων:



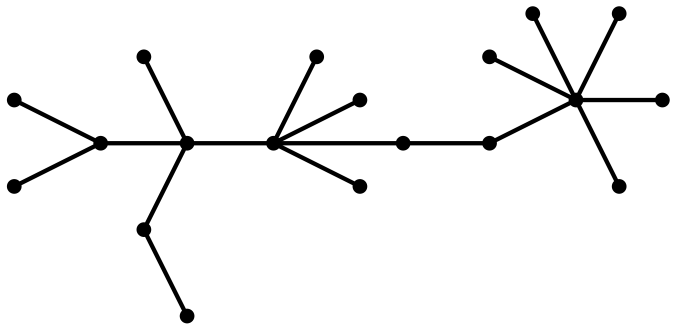
Για $n = 4$, έχουμε $\log_3(2 \cdot 4 + 1) = \log_3 9 = 2$, οπότε τα ζυγίσματα θα είναι τουλάχιστον 2, μια δε λύση με ακριβώς δύο ζυγίσματα φαίνεται στο παρακάτω 3-δένδρο αποφάσεων:



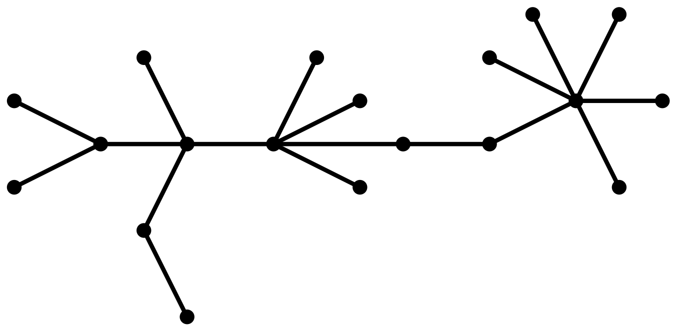
□

Ασκήσεις προς επίλυση

(1) Να βρεθεί το κέντρο του παρακάτω δένδρου.



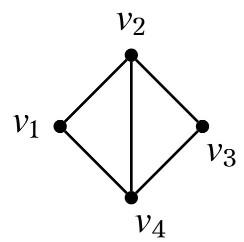
(2) Να βρεθεί το κεντροειδές του παρακάτω δένδρου.



(3) Να βρεθεί ο αριθμός των δένδρων ζεύξης των παρακάτω γραφημάτων

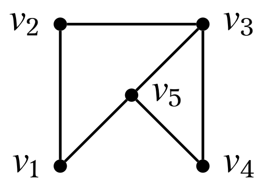
i)

G_1



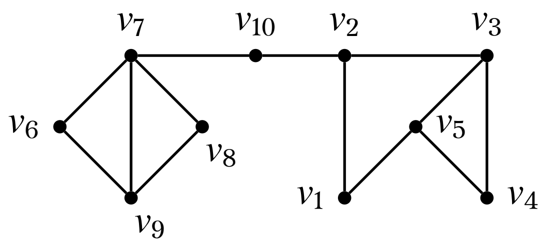
ii)

G_2

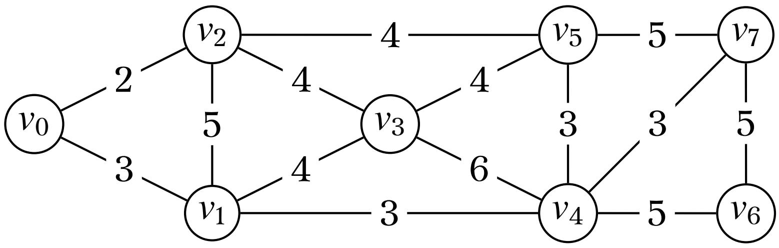


iii)

G_3



(4) Να βρεθεί ένα ελάχιστο δένδρο ζεύξης του γραφήματος



(5) Τι συνθήκη πρέπει να ικανοποιούν τα βάρη ενός γραφήματος ώστε το ελάχιστο δένδρο ζεύξης να είναι μοναδικό;

(6) Να λυθεί το πρόβλημα των κίβδηλων νομισμάτων για $n = 5$.

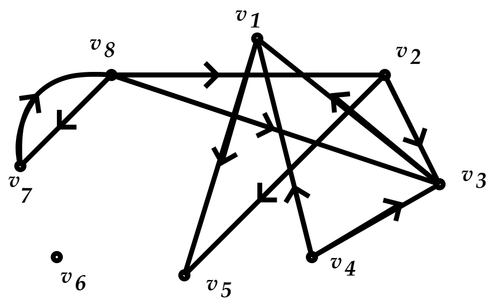
ΤΡΙΤΟ ΜΕΡΟΣ: ΓΡΑΦΗΜΑΤΑ ΤΟΞΩΝ

28. ΒΑΣΙΚΟΙ ΟΡΙΣΜΟΙ

Κάθε δυάδα $G = (V(G), U(G))$, ή (V, U), όπου V είναι ένα μη κενό σύνολο και U είναι ένα σύνολο από διατεταγμένα ζεύγη $(v, u) \in V^2$ ονομάζεται **γράφημα τόξων**, ή **προσανατολισμένο γράφημα** (directed graph), ή **γράφημα με κατεύθυνση** (oriented graph), ή **διγράφημα** (digraph).

Τα στοιχεία του V καλούνται **κορυφές**, ή **σημεία**, ή **κόμβοι** όπως και στα γραφήματα δεσμών, ενώ τα στοιχεία του U καλούνται **τόξα** (arcs) και συμβολίζονται γραφικά με τόξα.

Παράδειγμα: Η δυάδα $G = (V(G), U(G))$ όπου $V(G) = \{v_1, v_2, v_3, v_4,\ v_5, v_6, v_7, v_8\}$ και $U(G) = \{(v_1, v_5), (v_2, v_3), (v_2, v_5), (v_3, v_1), (v_4, v_1), (v_4, v_3),\ (v_7, v_8), (v_8, v_2), (v_8, v_3), (v_8, v_7)\}$ είναι ένα γράφημα τόξων. Η γραφική του απεικόνιση είναι η ακόλουθη:



Το τόξο (v, v), $v \in V$ ονομάζεται **βρόχος** (loop).

Οι ορισμοί είναι αντίστοιχοι με αυτούς που δώσαμε στα γραφήματα δεσμών με τις εξής επισημάνσεις:

Τώρα ορίζεται **βαθμός εξόδου** (out-degree) $d_+(v)$ ενός κόμβου v (πόσοι δεσμοί "φεύγουν" από τον κόμβο) και **βαθμός εισόδου** (in-degree) $d_-(v)$ (πόσοι δεσμοί "φθάνουν").

Έτσι,

$$d_+(v) = |\{u \in V(G) : (v, u) \in U(G)\}|,$$

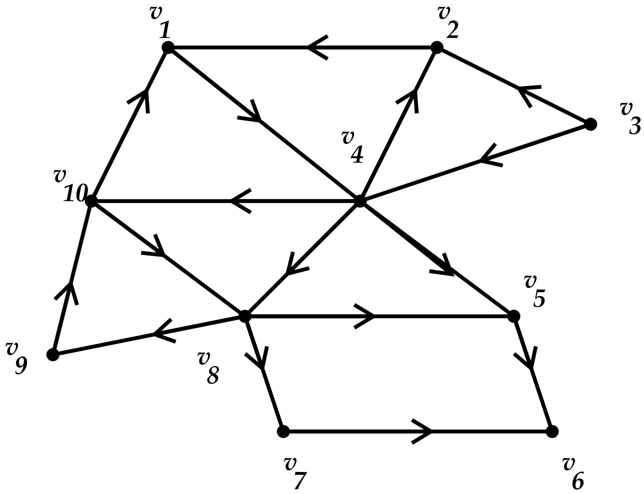
ενώ

$$d_-(v) = |\{u \in V(G) : (u, v) \in U(G)\}|$$

Προφανώς τώρα ο **βαθμός** $d(v)$ ενός κόμβου v ορίζεται από την σχέση

$$d(v) = d_+(v) + d_-(v).$$

Παράδειγμα: Στο επόμενο γράφημα



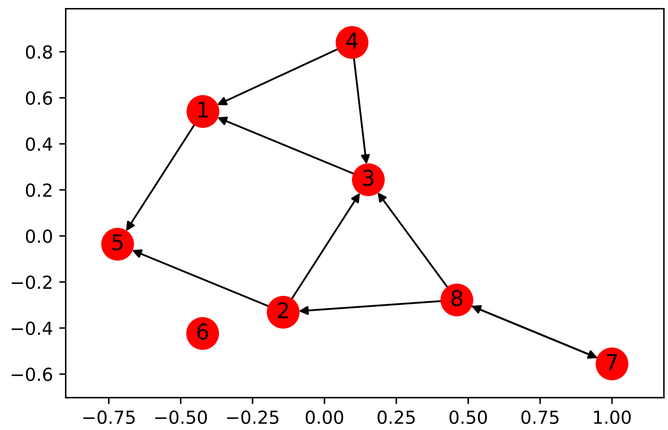
είναι $d_+(v_8) = 3$, $d_-(v_8) = 2$, $d(v_8) = 5$, $d_-(v_2) = 2$, $d_-(v_3) = 0$, κ.λπ.

Μπορούμε να ορίσουμε το γραφήματα τόξων του πρώτου παραδείγματος χρησιμοποιώντας την βιβλιοθήκη networkx:

```
import networkx as nx
import matplotlib.pyplot as plt

G = nx.DiGraph()
V = [1,2,3,4,5,6,7,8]
U = [(1,5),(2,3),(2,5),(3,1),(4,1),(4,3),(7,8),(8,2),(8,3),(8,7)]
G.add_nodes_from(V)
G.add_edges_from(U)

pos = nx.layout.kamada_kawai_layout(G)
nx.draw_networkx(G,pos)

for v in G:
    print("Vertex",v,"has indegree:",G.in_degree(v),"and outdegree:",G.out_degree(v))

plt.show()
```

Output:

```
Vertex 1 has indegree: 2 and outdegree: 1
Vertex 2 has indegree: 1 and outdegree: 2
Vertex 3 has indegree: 3 and outdegree: 1
Vertex 4 has indegree: 0 and outdegree: 2
Vertex 5 has indegree: 2 and outdegree: 0
Vertex 6 has indegree: 0 and outdegree: 0
Vertex 7 has indegree: 1 and outdegree: 1
Vertex 8 has indegree: 1 and outdegree: 3
```



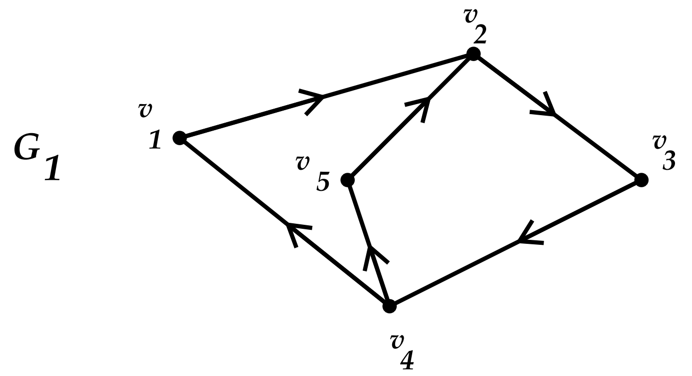
Η διαδρομή σε ένα διγράφημα πρέπει εν γένει να "ακολουθεί" τη διεύθυνση κάθε τόξου. Υπάρχουν διάφορα είδη συνεκτικότητας στα διγραφήματα:

Ένα γράφημα τόξων λέγεται **ισχυρά συνεκτικό** αν για οποιοδήποτε ζεύγος κόμβων του υπάρχει μονοπάτι και από τον πρώτο προς τον δεύτερο, και από τον δεύτερο προς τον πρώτο.

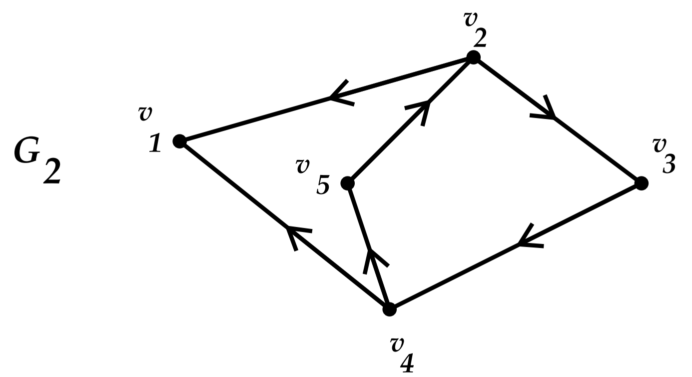
Ένα γράφημα τόξων λέγεται **μονομερώς συνεκτικό** αν δεν είναι ισχυρά συνεκτικό, αλλά για οποιοδήποτε ζεύγος κόμβων του υπάρχει μονοπάτι είτε από τον πρώτο προς τον δεύτερο, είτε από τον δεύτερο προς τον πρώτο.

Ένα γράφημα τόξων λέγεται **ασθενώς συνεκτικό** αν δεν είναι ισχυρά ή μονομερώς συνεκτικό, αλλά για οποιοδήποτε ζεύγος κόμβων του υπάρχει **ημιδιαδρομή** μεταξύ τους (δηλαδή τώρα επιτρέπεται και διάτρεξη τόξων αντίθετα με τον προσανατολισμό τους, όποτε χρειαστεί).

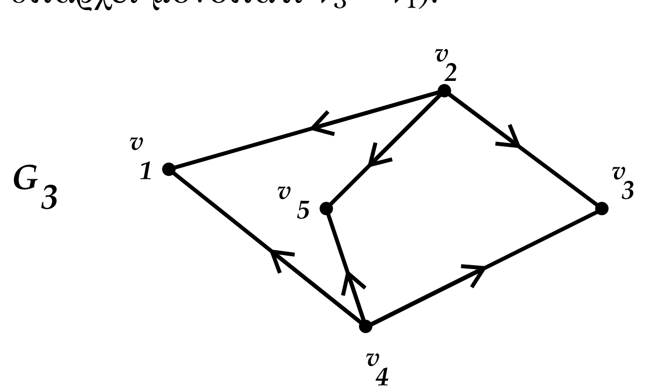
Παραδείγματα:



Το γράφημα G_1 είναι ισχυρά συνεκτικό.



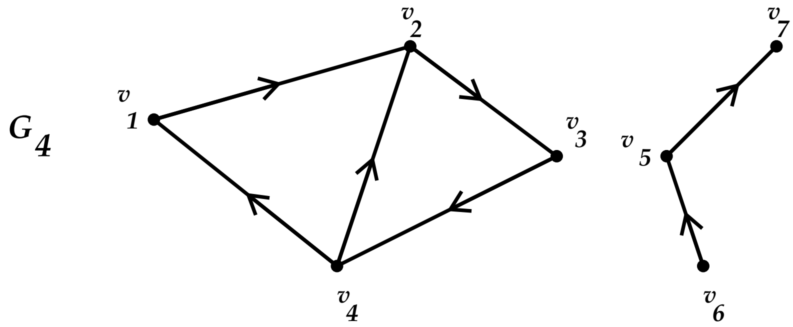
Το γράφημα G_2 είναι μονομερώς συνεκτικό (αφού για παράδειγμα, δεν υπάρχει $v_1 - v_3$ μονοπάτι, ενώ υπάρχει μονοπάτι $v_3 - v_1$).



Το γράφημα G_3 είναι ασθενώς συνεκτικό (αφού για παράδειγμα δεν υπάρχει ούτε $v_2 - v_4$, ούτε $v_4 - v_2$ μονοπάτι, ενώ υπάρχει η ημιδιαδρομή (v_2, v_3, v_4)).

Ένα γράφημα τόξων ονομάζεται **μη συνεκτικό** αν δεν είναι ούτε ασθενώς, ούτε μονομερώς, ούτε ισχυρά συνεκτικό.

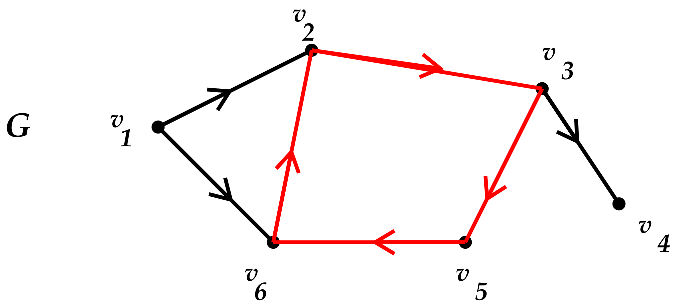
Παράδειγμα:



Το γράφημα G_4 είναι μη συνεκτικό.

Συνήθως ο κλειστός δρόμος που σχηματίζεται από τόξα λέγεται **κύκλωμα**.

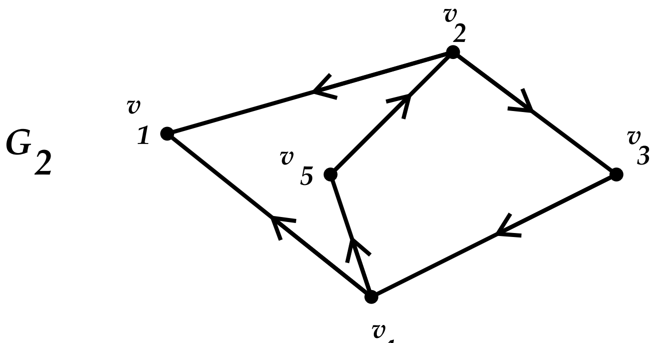
Παράδειγμα:



Στο γράφημα G, ο δρόμος $(v_2, v_3, v_5, v_6, v_2)$ είναι κύκλωμα, ενώ η ημιδιαδρομή (v_1, v_2, v_6, v_1) δεν είναι.

Ισχυρά συνεκτική συνιστώσα ενός γραφήματος τόξων G ονομάζεται οποιοδήποτε μεγιστικό ισχυρά συνεκτικό υπογράφημα του G.

Για παράδειγμα, το γράφημα G_2



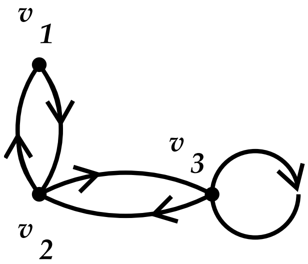
(το οποίο όπως είδαμε δεν είναι ισχυρά συνεκτικό) περιέχει μια ισχυρά συνεκτική συνιστώσα: το κύκλωμα $(v_2, v_3, v_4, v_5, v_2)$.

Στα γραφήματα τόξων ορίζουμε και τα παρακάτω είδη γραφημάτων:

Συμμετρικό ονομάζεται ένα γράφημα τόξων $G = (V, U)$ για το οποίο ισχύει

$$(u, v) \in U \Leftrightarrow (v, u) \in U.$$

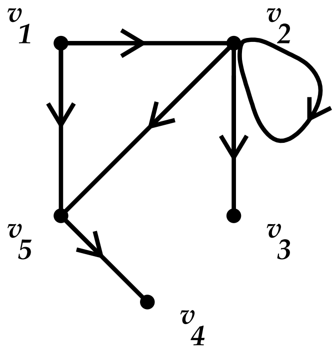
Παράδειγμα:



Αντισυμμετρικό ονομάζεται ένα γράφημα τόξων $G = (V, U)$ για το οποίο ισχύει

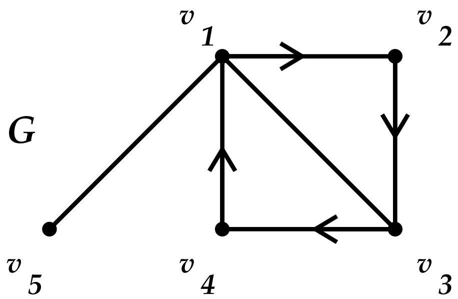
$$(u, v) \in U \text{ και } (v, u) \in U \Rightarrow u = v.$$

Παράδειγμα:

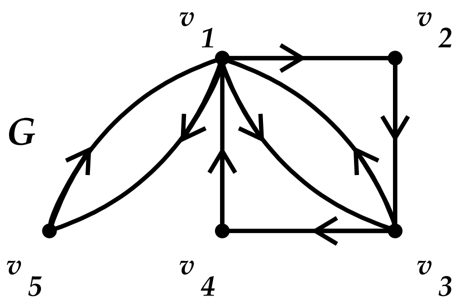


Παρατήρηση: Μερικές φορές εμφανίζονται γραφήματα που περιέχουν συγχρόνως και δεσμούς και τόξα.

Παράδειγμα:



Τα γραφήματα αυτά, τα θεωρούμε ουσιαστικά ως γραφήματα τόξων, αντικαθιστώντας κάθε δέσμο $\{v, u\}$ με δύο τόξα (v, u) και (u, v). Έτσι, το προηγούμενο παράδειγμα γράφεται:



Φυσικά, με την ίδια λογική μπορούμε γενικά οποιοδήποτε γράφημα δεσμών να το θεωρήσουμε αντίστοιχα ως γράφημα τόξων, το οποίο θα είναι προφανώς συμμετρικό. Το μειονέκτημα μιας τέτοιας προσέγγισης είναι ότι η αντίστοιχη θεωρία και οι εφαρμογές γίνονται γενικά πολύ πιο πολύπλοκες.

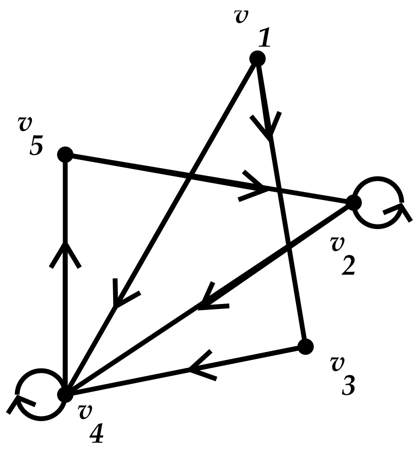
29. ΜΗΤΡΑ ΓΡΑΦΗΜΑΤΟΣ ΤΟΞΩΝ

Έστω $G = (V, U)$ ένα γράφημα τόξων. Ορίζουμε την $|V| \times |V|$ μήτρα M_G ή M του G ως εξής:

$$M = [m_{ij}], \text{ με } m_{ij} = \begin{cases} 1, & \text{αν } (v_i, v_j) \in U \\ 0, & \text{αν } (v_i, v_j) \notin U. \end{cases}$$

Η μήτρα αυτή ονομάζεται **μήτρα (γειτονικότητας) του γραφήματος τόξων**.

Παράδειγμα: Στο γράφημα G



αντιστοιχεί η μήτρα

$$M = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Ισχύει η παρακάτω πρόταση.

Πρόταση 48. *Ο αριθμός των v_i-v_j διαδρομών μήκους ν σε ένα γράφημα τόξων G, ισούται με το στοιχείο m_{ij} της μήτρας $M^\nu = [m_{ij}]$.*

Απόδειξη. Για $\nu = 1$ προφανώς ισχύει.

Έστω ότι ισχύει για $\nu = k$ (και έστω ότι $M = [m_{ij}]$ και $M^k = [p_{ij}]$).

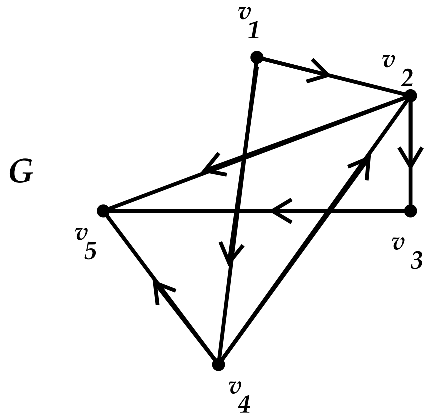
Για $\nu = k + 1$ θα είναι $M^{k+1} = M^k M = [q_{ij}]$, όπου

$$q_{ij} = \sum_{r=1}^{n} p_{ir} m_{rj} \tag{1}$$

(όπου $n = |V|$).

Από την υπόθεση της επαγωγής, p_{ir} είναι ο αριθμός των διαδρομών μήκους k από τον v_i στον v_r. Επίσης, m_{rj} είναι ο αριθμός των τόξων (δηλαδή των διαδρομών μήκους 1) από τον v_r στον v_j. Τότε το $p_{ir} m_{rj}$ είναι ο αριθμός των διαδρομών μήκους $k + 1$ από τον v_i στον v_j, με προτελευταίο κόμβο τον v_r. Άρα το $\sum_{r=1}^{n} p_{ir} m_{rj}$ (δηλαδή, λόγω της (1), το q_{ij}) θα είναι ο αριθμός όλων των διαδρομών μήκους $k + 1$ από τον v_i στον v_j με προτελευταίο κόμβο έναν από τους $v_1, v_2, \ldots, v_n$, δηλαδή θα είναι πράγματι ο αριθμός όλων των διαδρομών μήκους $k + 1$ από τον v_i στον v_j. □

Παράδειγμα: Για το γράφημα



έχουμε

$$M = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \qquad M^2 = \begin{bmatrix} 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$M^3 = \begin{bmatrix} 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \qquad M^4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \qquad M^5 = O_5.$$

Στην M^3 έχουμε $q_{15} = 2$, άρα υπάρχουν δύο διαδρομές μήκους 3 από το v_1 ως το v_5. (Πράγματι, είναι οι (v_1, v_2, v_3, v_5), (v_1, v_4, v_2, v_5)), ενώ $q_{14} = 0$, άρα δεν υπάρχει διαδρομή μήκους 3 από το v_1 στο v_4. Στην M^4 έχουμε $q_{15} = 1$, άρα υπάρχει μια διαδρομή μήκους 4 από το v_1 ως το v_5: $(v_1, v_4, v_2, v_3, v_5)$.

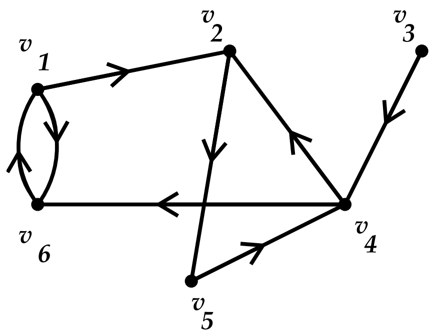
30. ΑΠΕΙΚΟΝΙΣΗ ΓΡΑΦΗΜΑΤΟΣ ΤΟΞΩΝ

Έστω $G = (V, U)$. Ορίζουμε την απεικόνιση

$$\Gamma : V \to \mathcal{P}(V) \text{ με } \Gamma(v) = \{u \in V : (v, u) \in U\}.$$

Παρατήρηση: Το ζεύγος (V, Γ) ορίζει το γράφημα G ισοδύναμα με το (V, U) και γι' αυτό μπορούμε να αναφερόμαστε και στο γράφημα (V, Γ) αντί (V, U). Η Γ ονομάζεται **απεικόνιση του γραφήματος τόξων**.

Παράδειγμα: Για το γράφημα



έχουμε $\Gamma(v_1) = \{v_2, v_6\}$, $\Gamma(v_2) = \{v_5\}$, $\Gamma(v_3) = \{v_4\}$ κ.λπ.

Αντίστοιχα με τα γραφήματα δεσμών, αν $A = \{v_1, v_2, \ldots, v_k\}$, τότε

$$\Gamma(A) = \Gamma(v_1) \cup \Gamma(v_2) \cup \cdots \cup \Gamma(v_k)$$

και (αναδρομικά), για $n \in \mathbb{N}^*$:

$$\Gamma^n(v) = \Gamma(\Gamma^{n-1}(v)).$$

Παραδείγματα: Για το τελευταίο γράφημα έχουμε:
$\Gamma^2(v_1) = \Gamma(\Gamma(v_1)) = \Gamma(\{v_2, v_6\}) = \{v_1, v_5\}$ και
$\Gamma^3(v_1) = \Gamma(\Gamma^2(v_1)) = \Gamma(\{v_1, v_5\}) = \{v_2, v_4, v_6\}$.

Ανάλογα ορίζουμε

$$\Gamma^{-1} : V \to \mathcal{P}(V) \text{ με } \Gamma^{-1}(v) = \{u \in V : (u, v) \in U\},$$

$$\Gamma^{-1}(A) = \Gamma^{-1}(v_1) \cup \Gamma^{-1}(v_2) \cup \cdots \cup \Gamma^{-1}(v_k), \text{ όπου } A = \{v_1, v_2, \ldots, v_k\}$$

και (αναδρομικά) για $n \in \mathbb{N}^*$:

$$\Gamma^{-n}(v) = \Gamma^{-1}(\Gamma^{-n+1}(v)).$$

Παράδειγμα: Για το τελευταίο γράφημα έχουμε:
$\Gamma^{-1}(v_1) = \{v_6\}$, $\Gamma^{-1}(v_2) = \{v_1, v_4\}$, $\Gamma^{-1}(v_3) = \emptyset$ κ.λπ. και
$\Gamma^{-2}(v_6) = \Gamma^{-1}(\Gamma^{-1}(v_6)) = \{v_3, v_5, v_6\}$.

Τέλος ορίζουμε τη **μεταβατική πρόσβαση** $\widehat{\Gamma}$ ενός κόμβου $v \in V$ ως εξής:

$$\widehat{\Gamma}(v) = \{v\} \cup \Gamma(v) \cup \Gamma^2(v) \cup \cdots$$

Παράδειγμα: Για το τελευταίο γράφημα έχουμε:
$\widehat{\Gamma}(v_4) = \{v_1, v_2, v_4, v_5, v_6\}$,
$\widehat{\Gamma}(v_3) = V$, κ.λπ.

Ισχύουν τα παρακάτω αποτελέσματα:

Πρόταση 49. *Το υποσύνολο των κόμβων ενός γραφήματος τόξων στους οποίους καταλήγουν οι διαδρομές που αρχίζουν από τον κόμβο* v *και έχουν μήκος* k*, ισούται με* $\Gamma^k(v)$.

Η απόδειξη γίνεται με επαγωγή.

Πρόταση 50. *Το υποσύνολο των κόμβων ενός γραφήματος τόξων από τους οποίους αρχίζουν οι διαδρομές μήκους* k *που καταλήγουν στον κόμβο* v*, ισούται με* $\Gamma^{-k}(v)$.

Η απόδειξη γίνεται με επαγωγή.

Πρόταση 51. *Το υποσύνολο των κόμβων ενός γραφήματος τόξων στους οποίους καταλήγουν οι διαδρομές που αρχίζουν από τον κόμβο* v *ισούται με* $\widehat{\Gamma}(v)$.

Η απόδειξη είναι προφανής.

Πρόταση 52. *Αν* $\widehat{\Gamma}(v) = V$*, για κάθε* $v \in V$*, τότε από κάθε κόμβο αρχίζουν διαδρομές που καταλήγουν σε κάθε άλλο κόμβο του γραφήματος.*

Η απόδειξη είναι προφανής.

31. ΠΥΡΗΝΑΣ - ΒΑΣΕΙΣ

Έστω $G = (V, \Gamma)$ ένα γράφημα τόξων.

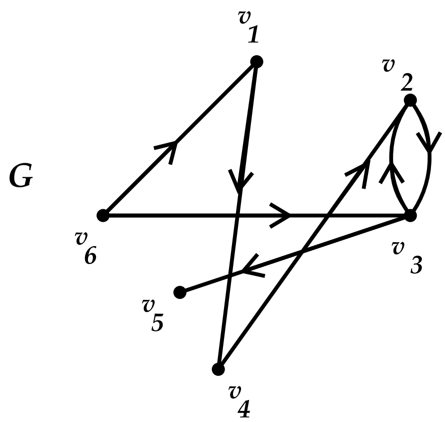
Το $S \subseteq V$ λέγεται **ευσταθές** (stable) σύνολο, όταν δυο οποιοιδήποτε κόμβοι του δεν συνδέονται με τόξο, δηλαδή όταν

$$v \in S \Rightarrow \Gamma(v) \cap S = \emptyset.$$

Το $A \subseteq V$ λέγεται **απορροφητικό** (absorbing) σύνολο, όταν κάθε κόμβος που δεν ανήκει στο A συνδέεται με τόξο, με ένα τουλάχιστον κόμβο προς το A, δηλαδή όταν

$$v \notin A \Rightarrow \Gamma(v) \cap A \neq \emptyset.$$

Παράδειγμα: Για το γράφημα G

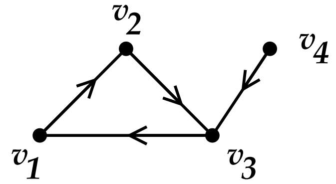


το σύνολο $\{v_4, v_5, v_6\}$ είναι ευσταθές σύνολο, το σύνολο $\{v_3, v_4, v_5\}$ είναι απορροφητικό σύνολο, ενώ το σύνολο $\{v_1, v_2, v_5\}$ είναι συγχρόνως ευσταθές και απορροφητικό.

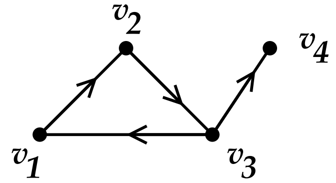
Το $N \subseteq V$ λέγεται **πυρήνας** (kernel) όταν είναι συγχρόνως και ευσταθές και απορροφητικό.

Ένα γράφημα τόξων μπορεί να μην έχει πυρήνα.

Παραδείγματα:



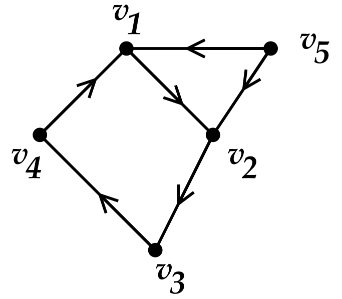
Δεν έχει πυρήνα.



Το σύνολο $\{v_2, v_4\}$ είναι ένας πυρήνας.

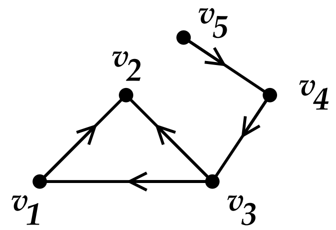
Πρόταση 53. *Ένα γράφημα τόξων που δεν έχει κυκλώματα περιττού μήκους έχει τουλάχιστον ένα πυρήνα.*

Παράδειγμα:



Τα σύνολα $\{v_1, v_3\}$ και $\{v_2, v_4\}$ είναι δύο πυρήνες του γραφήματος.

Πρόταση 54. *Ένα γράφημα τόξων χωρίς κυκλώματα έχει ένα μόνο πυρήνα.*



Το σύνολο $\{v_2, v_4\}$ είναι ο μοναδικός πυρήνας του γραφήματος.

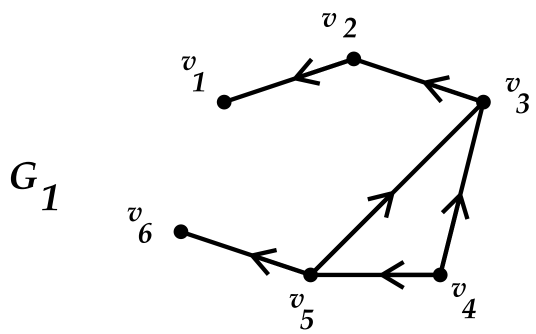
Βάση του $G = (V, \Gamma)$ ονομάζεται κάθε $B \subset V$ για το οποίο:
i) Δεν υπάρχει δρόμος που να συνδέει δύο οποιουσδήποτε κόμβους του.
ii) Κάθε κόμβος $x \notin B$ είναι η αρχή δρόμου με πέρας κόμβο του B.
(Αν $|B| = 1$, τότε το B ονομάζεται **ρίζα** του G).

Αντιβάση ονομάζεται κάθε $T \subset V$ για το οποίο:
i) Δεν υπάρχει δρόμος που να συνδέει δύο οποιουσδήποτε κόμβους του.
ii) Κάθε κόμβος $x \notin T$ είναι πέρας δρόμου με αρχή κόμβο του T.
(Αν $|T| = 1$, τότε το T ονομάζεται **αντιρίζα** του G).

(Στα δίκτυα επικοινωνιών οι αντιβάσεις είναι οι πηγές πληροφοριών, ενώ οι βάσεις είναι οι τελικοί αποδέκτες).

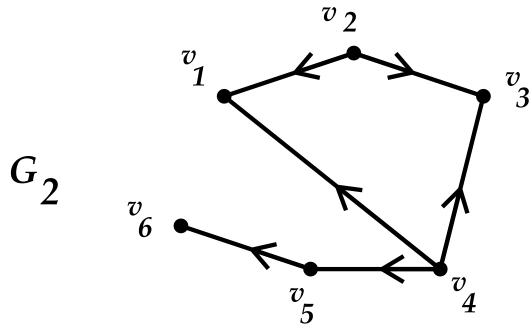
Παραδείγματα:

1. Μια βάση του γραφήματος G_1



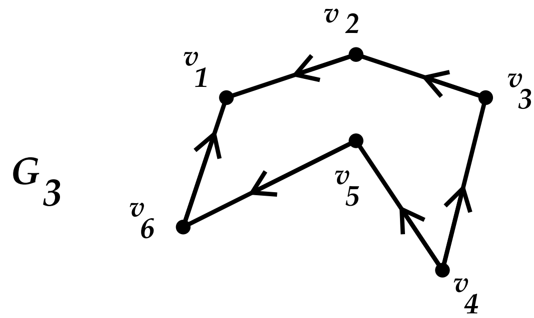
είναι η $B = \{v_1, v_6\}$. Το γράφημα G_1 δεν έχει ρίζα.

2. Μια αντιβάση του γραφήματος G_2



είναι η $T = \{v_2, v_4\}$. Το γράφημα G_2 δεν έχει αντιρίζα.

3. Στο γράφημα G_3 το v_1 είναι ρίζα ενώ το v_4 είναι αντιρίζα.



4. Έστω V το σύνολο των προτάσεων μιας θεωρίας η οποία αποτελείται από αξιώματα και θεωρήματα, με U το σύνολο των τόξων που ορίζονται ως εξής:

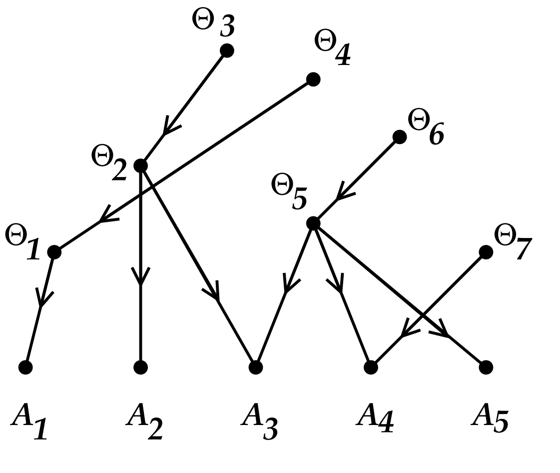
$$(\alpha, \beta) \in U \text{ ανν το } \beta \text{ χρησιμοποιείται για την απόδειξη του } \alpha.$$

Δεδομένου ότι στο σύνολο των προτάσεων μιας θεωρίας

i) Κανένα αξίωμά της δεν προκύπτει από άλλο αξίωμα

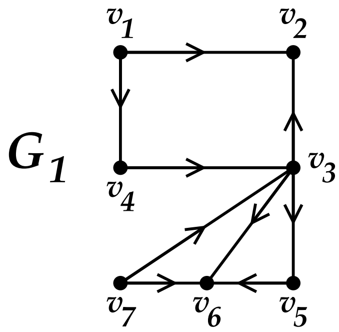
ii) Κάθε πρόταση προκύπτει από ένα τουλάχιστον αξίωμα,

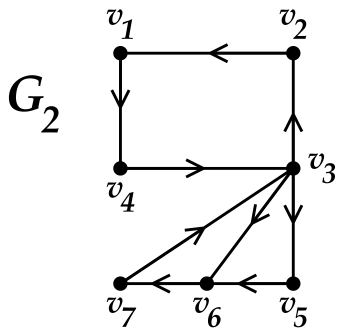
το σύνολο των αξιωμάτων μιας θεωρίας πρέπει να σχηματίζει μια βάση.

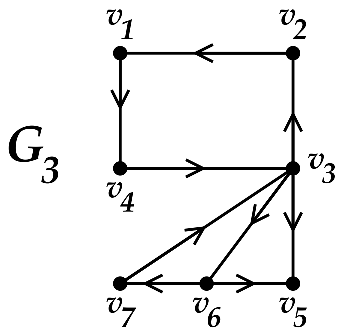


Ασκήσεις προς επίλυση

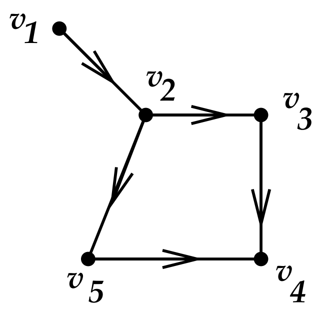
(1) Να βρεθεί για καθένα από τα παρακάτω γραφήματα τόξων αν είναι ισχυρά, μονομερώς, ή ασθενώς συνεκτικό.







(2) Δίδεται το γράφημα



Να βρεθεί με τη βοήθεια της μήτρας του γραφήματος:

α) Πόσες διαδρομές μήκους 2 υπάρχουν ανάμεσα στις κορυφές v_1, v_3,

β) Πόσες διαδρομές μήκους 2 υπάρχουν ανάμεσα στις κορυφές v_1, v_5,

γ) Πόσες διαδρομές μήκους 2 υπάρχουν ανάμεσα στις κορυφές v_2, v_3,

δ) Πόσες διαδρομές μήκους 3 υπάρχουν ανάμεσα στις κορυφές v_2, v_4,

ε) Πόσες διαδρομές μήκους 3 υπάρχουν ανάμεσα στις κορυφές v_1, v_4.

Σε κάθε μία από τις περιπτώσεις αυτές να προσδιοριστούν οι αντίστοιχες διαδρομές.

Μήτρες για την απάντηση στην άσκηση:

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad M^2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad M^3 = \begin{bmatrix} 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

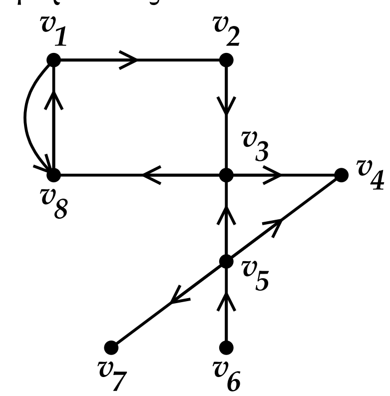
(3) Δίδεται το γράφημα τόξων $G = (X, U)$, όπου
$X = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$,
$U = \{(v_1, v_2),\ (v_1, v_8),\ (v_2, v_1),\ (v_2, v_3),\ (v_2, v_5),\ (v_2, v_7),\ (v_2, v_8),\ (v_3, v_4),\ (v_4, v_3),\ (v_4, v_5),\ (v_8, v_1)\}$.

i) Να περιγραφεί αυτό το γράφημα με τρεις άλλους ισοδύναμους τρόπους.

ii) Να υπολογισθούν τα $d_+(v_2)$, $d_-(v_2)$, $d_+(v_4)$, $d(v_8)$.

iii) Να ευρεθούν τα $\Gamma(v_1)$, $\Gamma(v_5)$, $\Gamma^{-1}(v_5)$, $\Gamma^2(v_1)$, $\widehat{\Gamma}(v_1)$, $\widehat{\Gamma}(v_3)$.

(4) Δίδεται το παρακάτω γράφημα τόξων:



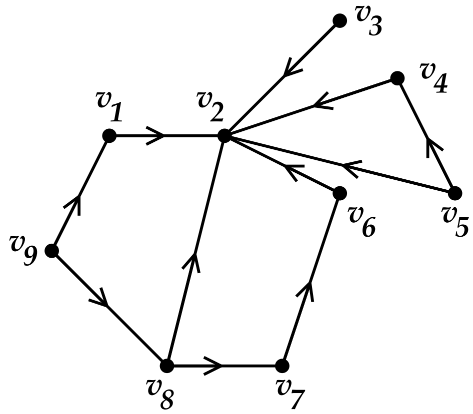
Να ευρεθεί, αν υπάρχει:

i) Ένα ευσταθές σύνολο κόμβων.

ii) Ένα απορροφητικό σύνολο κόμβων.

iii) Ένας πυρήνας του.

(5) Δίδεται το παρακάτω γράφημα τόξων:



Να ευρεθεί, αν υπάρχει:

i) Μια βάση του.

ii) Μια ρίζα του.

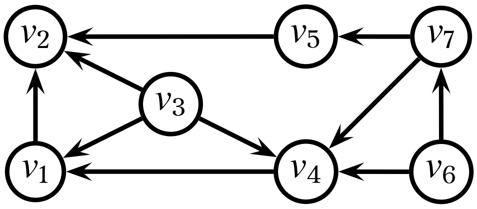
iii) Μια αντιβάση του.

iv) Μια αντιρίζα του.

ΤΕΤΑΡΤΟ ΜΕΡΟΣ: ΕΦΑΡΜΟΓΕΣ - ΠΑΡΑΔΕΙΓΜΑΤΑ

32. ΤΟΠΟΛΟΓΙΚΗ ΔΙΑΤΑΞΗ

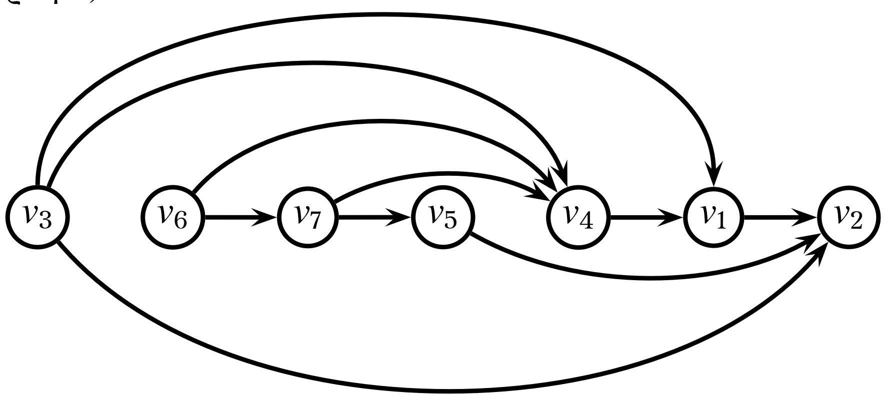
Έστω $G = (V, U)$ ένα γράφημα τόξων. Μια ολική διάταξη $<$ στο σύνολο των κορυφών του G ονομάζεται **τοπολογική** (topological sorting) αν και μόνο αν για κάθε τόξο $(v, u) \in U$ ισχύει ότι $v < u$ (v προηγείται του u) στην διάταξη.
Παράδειγμα: Στο επόμενο γράφημα τόξων



μια τοπολογική διάταξη κορυφών του είναι η διάταξη:

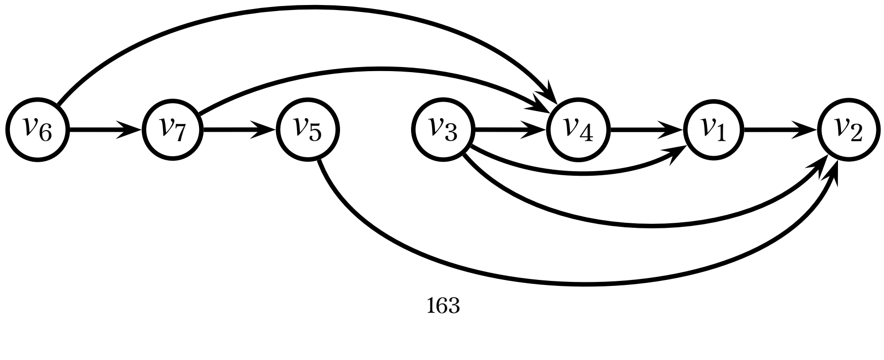
$$(v_3,\ v_6, v_7,\ v_5,\ v_4,\ v_1,\ v_2)$$

Μπορούμε εύκολα να επιβεβαιώσουμε ότι η σειρά αυτή ικανοποιεί τον ορισμό της τοπολογικής διάταξης σχεδιάζοντας ξανά το γράφημα, τοποθετώντας τις κορυφές του πάνω σε μια ευθεία με την σειρά που εμφανίζονται στην διάταξη. Τότε όλα τα τόξα έχουν προσανατολισμό από τα αριστερά προς τα δεξιά (από μικρότερη προς μεγαλύτερη κορυφή).

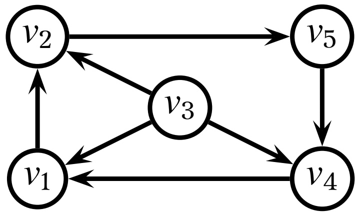


Μια άλλη τοπολογική διάταξη είναι η διάταξη:

$$(v_6,\ v_7, v_5,\ v_3,\ v_4,\ v_1,\ v_2)$$



Το επόμενο γράφημα τόξων δεν έχει τοπολογική διάταξη



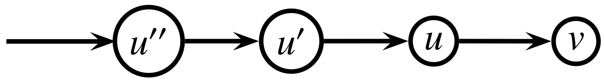
διότι η ύπαρξη του κυκλώματος (v_1, v_2, v_5, v_4, v_1) παραβιάζει την μεταβατική ιδιότητα που έχουν όλες οι διατάξεις.

Πρόταση 55. *Ένα γράφημα τόξων G έχει τοπολογική διάταξη αν και μόνο αν δεν περιέχει κυκλώματα.*

Παρατήρηση: Σε κάθε γράφημα τόξων χωρίς κυκλώματα υπάρχει τουλάχιστον μια κορυφή με βαθμό εισόδου 0.

Πράγματι, έστω ότι δεν υπάρχει κορυφή με βαθμό εισόδου 0, τότε για κάθε κορυφή v υπάρχει προηγούμενη κορυφή u τέτοια ώστε $(u, v) \in U$.

Οπότε, αν ξεκινήσουμε από μια οποιαδήποτε κορυφή v μπορούμε να δημιουργήσουμε μια λίστα που κατασκευάζεται προθέτοντας την προηγούμενη κορυφή u' κάθε κορυφής u που είναι στην λίστα.



Επειδή το πλήθος των κορυφών είναι φραγμένο, η λίστα αυτή θα περιέχει επαναλήψεις, επομένως το γράφημα θα έχει κύκλωμα, άτοπο.

Η εύρεση μιας τοπολογικής διάταξης των κορυφών ενός γραφήματος τόξων G, αν υπάρχει, μπορεί να γίνει μέσω του επόμενου αλγόριθμου.

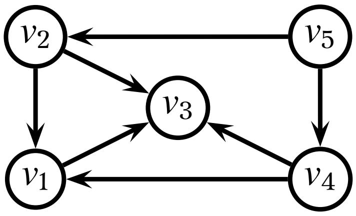
Αλγόριθμος τοπολογικής διάταξης

Είσοδος: Ένα γράφημα τόξων G

Έξοδος: Μια λίστα L με την τοπολογική διάταξη των κορυφών, αν υπάρχει.

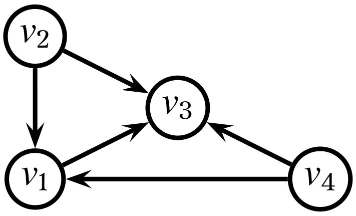
- Όσο υπάρχουν κορυφές με βαθμό εισόδου 0 στο G
  - Επιλέγουμε μια κορυφή v του G με βαθμό εισόδου 0 και την τοποθετούμε στο τέλος της λίστας L.
  - Αφαιρούμε από το G την κορυφή v (καθώς και όλα τα τόξα που ξεκινούν από αυτήν).
- Αν το G δεν περιέχει άλλες κορυφές, τότε η σειρά των κορυφών στην λίστα L είναι μια τοπολογική διάταξη.
- Αλλιώς, το γράφημα περιέχει κύκλωμα και άρα δεν έχει τοπολογική διάταξη.

Παράδειγμα Να βρεθεί, αν υπάρχει, μια τοπολογική διάταξη των κορυφών του γραφήματος:



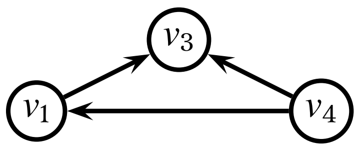
Λύση.

- Η μοναδική κορυφή με βαθμό εισόδου 0 είναι η v_5. Τοποθετούμε την v_5 στο τέλος της λίστας L και αφαιρούμε τα τόξα που αρχίζουν από την v_5



$$L = (v_5)$$

- Μετά την αφαίρεση της v_5 υπάρχουν δύο κορυφές με βαθμό εισόδου 0, οι v_2 και v_4. Επιλέγουμε την v_2, την τοποθετούμε στο τέλος της λίστας L, και αφαιρούμε όλα τα τόξα της



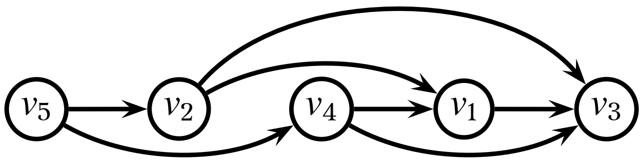
$$L = (v_5, v_2)$$

- Συνεχίζοντας με τον ίδιο τρόπο καταλήγουμε στο κενό γράφημα και στην λίστα

$$L = (v_5, v_2, v_4, v_1, v_3)$$

η οποία είναι τοπολογική διάταξη των κορυφών του γραφήματος.

Πράγματι,



□

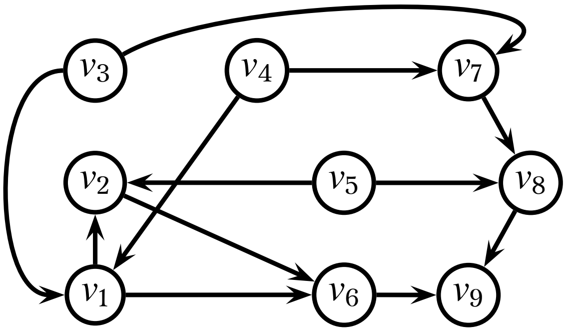
Παρατήρηση: Για την εύρεση της τοπολογικής διάταξης ενός γραφήματος υπάρχει και άλλος αλγόριθμος που χρησιμοποιεί την αναζήτηση σε βάθος των κορυφών του.

Άσκηση 12. *Για την ολοκλήρωση ενός έργου πρέπει να εκτελεστούν 9 δραστηριότητες* $T_1, T_2, \ldots, T_9$. *Κάποιες από αυτές χρειάζονται τα αποτελέσματα μερικών άλλων, των οποίων η εκτέλεση πρέπει να προηγηθεί. Οι απαιτήσεις κάθε μιας δίδονται στον επόμενο πίνακα:*

	απαιτήσεις		*απαιτήσεις*		*απαιτήσεις*
T_1	T_3, T_4	T_4		T_7	T_3, T_4
T_2	T_1, T_5	T_5		T_8	T_5, T_7
T_3		T_6	T_1, T_2	T_9	T_6, T_8

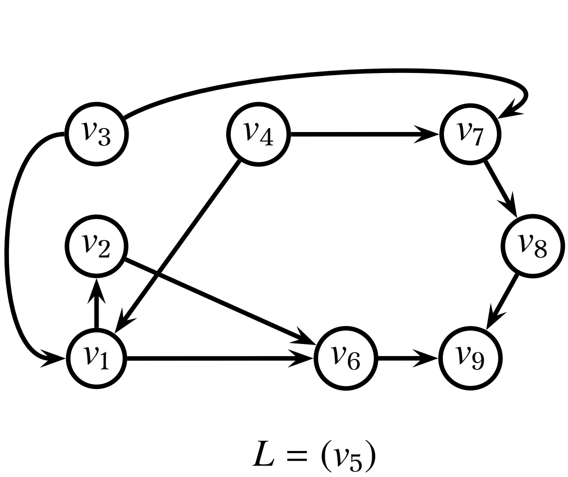
Να βρεθεί με ποια σειρά πρέπει να εκτελεστούν οι $T_1, T_2, \ldots, T_9$ *ώστε να ολοκληρωθεί το έργο.*

Λύση. Στο πρόβλημα της εκτέλεσης αντιστοιχεί ένα κατευθυνόμενο γράφημα στο οποίο η δραστηριότητα T_i αναπαρίσταται από την κορυφή v_i, και αν η δραστηριότητα T_i απαιτεί την ολοκλήρωση της δραστηριότητας T_j τότε στην κορυφή v_i καταλήγει ένα τόξο με αρχή την κορυφή v_j.



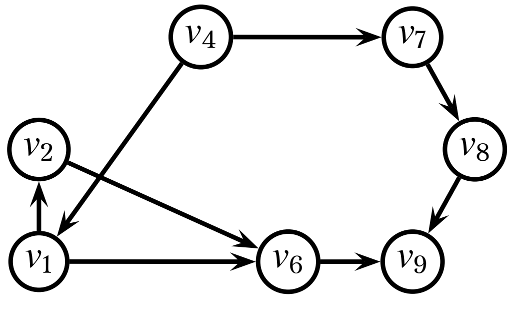
Το πρόβλημα είναι ισοδύναμο με την εύρεση της τοπολογικής διάταξης των κορυφών του γραφήματος. Προκειμένου να υπάρχει λύση πρέπει να μην υπάρχει κύκλωμα. Θα εφαρμόσουμε τον αλγόριθμο της τοπολογικής διάταξης. Θα φτιάξουμε μια λίστα L που θα περιέχει τις κορυφές του γραφήματος με την σειρά της τοπολογικής διάταξης. Κάθε φορά επιλέγουμε μια κορυφή με βαθμό εισόδου 0, την προσθέτουμε στο τέλος της λίστας L και αφαιρούμε όλα τα τόξα που αρχίζουν από αυτή, μέχρις ότου να μην υπάρχουν κορυφές με βαθμό εισόδου 0 στο γράφημα.

1. Επιλέγουμε την v_5 οπότε έχουμε:



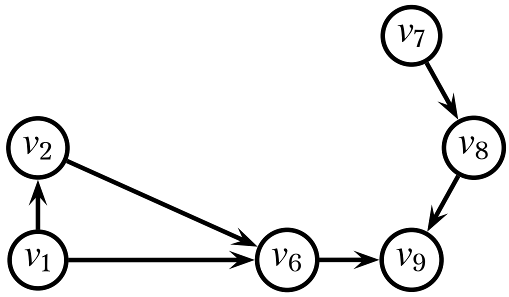
$L = (v_5)$

2. Επιλέγουμε την v_3 οπότε έχουμε:



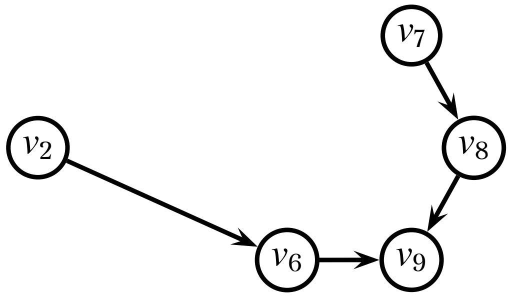
$L = (v_5, v_3)$

3. Επιλέγουμε την v_4 οπότε έχουμε:



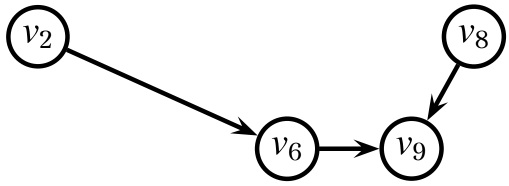
$$L = (v_5, v_3, v_4)$$

4. Επιλέγουμε την v_1 οπότε έχουμε:



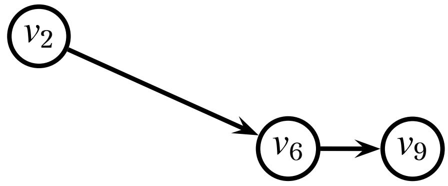
$$L = (v_5, v_3, v_4, v_1)$$

5. Επιλέγουμε την v_7 οπότε έχουμε:



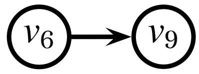
$$L = (v_5, v_3, v_4, v_1, v_7)$$

6. Επιλέγουμε την v_8 οπότε έχουμε:



$$L = (v_5, v_3, v_4, v_1, v_7, v_8)$$

7. Επιλέγουμε την v_2 οπότε έχουμε:



$$L = (v_5, v_3, v_4, v_1, v_7, v_8, v_2)$$

8. Επιλέγουμε την v_6 οπότε έχουμε:

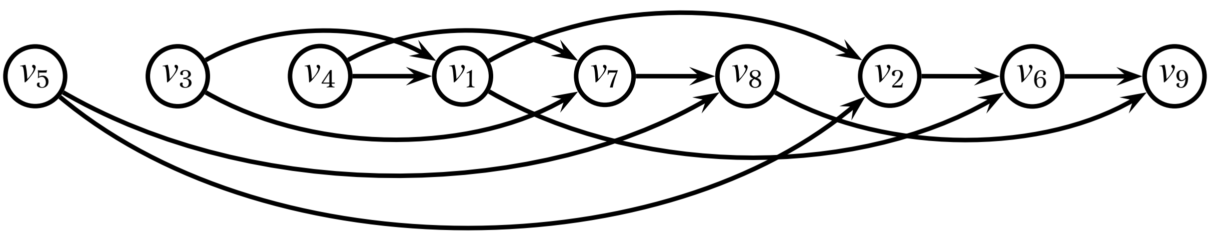
$$L = (v_5, v_3, v_4, v_1, v_7, v_8, v_2, v_6)$$

9. Επιλέγουμε την v_9 οπότε έχουμε:

$$L = (v_5, v_3, v_4, v_1, v_7, v_8, v_2, v_6, v_9)$$

Μια τοπολογική διάταξη των κορυφών του γραφήματος είναι η σειρά

$$(v_5, v_3, v_4, v_1, v_7, v_8, v_2, v_6, v_9)$$



Επομένως, μια πιθανή σειρά εκτέλεσης των 9 δραστηριοτήτων έτσι ώστε να ικανοποιούνται οι απαιτήσεις της καθεμιάς είναι η εξής:

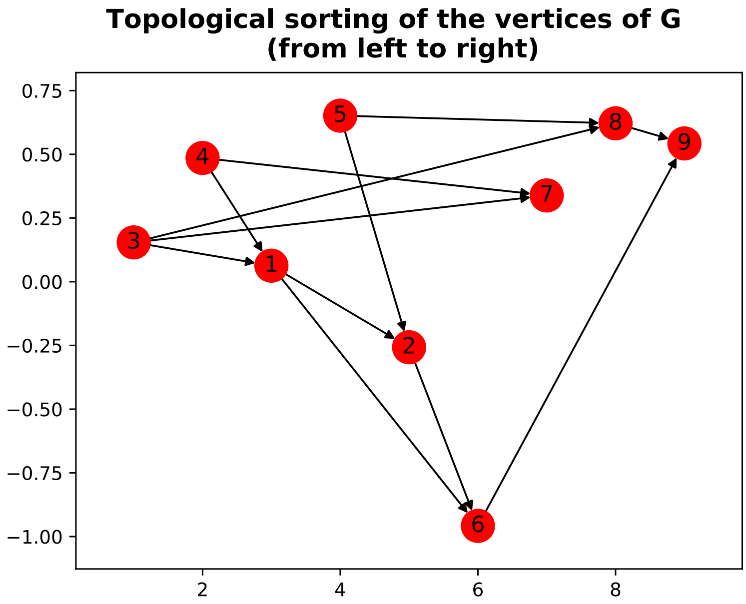
$$(T_5, T_3, T_4, T_1, T_7, T_8, T_2, T_6, T_9)$$

Μπορούμε να υπολογίζουμε την τοπολογική διάταξη του G χρησιμοποιώντας την βιβλιοθήκη networkx:

```
import networkx as nx
import matplotlib.pyplot as plt
import random

G = nx.DiGraph()
V = [1,2,3,4,5,6,7,8,9]
U = [(1,2),(1,6),(2,6),(3,1),(3,7),(4,1),(4,7),(5,2),(5,8),(6,9),(3,8),(8,9)]
G.add_nodes_from(V)
G.add_edges_from(U)

# Topological sorting of G using method topological_sort(G)
if(nx.is_directed_acyclic_graph(G)):
    R = list(nx.topological_sort(G))
    print("A topological sorting of G:",R,"(using the method topological_sort())")
else:
    print("G contains a cycle")

# A simple implementation of topological sorting algorithm
L = [] #Topological sorting of the nodes of G
H = G.copy() #Work on an copy of G
while(len(H)!=0):
    notfound = True
    for v in H.nodes():
        if (H.in_degree(v) == 0):
           L.append(v)
           H.remove_node(v)
           notfound = False # vertex v has indegree 0
           break
    if notfound: break # no vertex has indegree 0
# If L contains all the nodes of G a solution is found
if len(L) == len(G):
    print("A topological sorting of G:",L)
else:
    print("G contains a cycle")

# Position all vertices of G in a line according to the topological sorting L
x = y = 0
mypos = {} #empty dictionary
for i in L:
    x = x + 1
    y = random.uniform(-1,1)
    mypos[i] = [x,y] #coordinates of node i

# Create a figure in order to add title text
fig = plt.figure()
fig.suptitle('Topological sorting of the vertices of G \n (from left to right)',
   fontsize=14, fontweight='bold')
nx.draw_networkx(G,pos=mypos)
plt.show()
```

Output:

```
A topological sorting of G: [5, 4, 3, 8, 7, 1, 2, 6, 9] (using the method
   topological_sort())
A topological sorting of G: [3, 4, 1, 5, 2, 6, 7, 8, 9]
```



□

Παρατήρηση: Για την εύρεση της τοπολογικής διάταξης ενός γραφήματος υπάρχει και άλλος αλγόριθμος που χρησιμοποιεί την αναζήτηση σε βάθος των κορυφών του.

33. ΣΥΝΑΡΤΗΣΗ GRUNDY - SPRAGUE

Έστω $A \subseteq \mathbb{N}$. Ορίζουμε το mex **(minimum excluded value)** του A ως εξής:

$$\operatorname{mex} A = \min \mathbb{N} \setminus A,$$

δηλαδή $\operatorname{mex} A$ **είναι ο ελάχιστος φυσικός αριθμός που δεν ανήκει στο** A.
Παράδειγμα:

- $\operatorname{mex}\{1, 2, 4\} = 0$
- $\operatorname{mex}\{0, 1, 2, 6\} = 3$
- $\operatorname{mex}\{0, 1, 2, 3\} = 4$
- $\operatorname{mex} \emptyset = 0.$

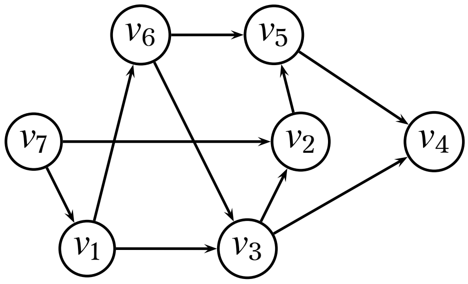
Για κάθε κορυφή v ενός γραφήματος τόξων $G = (V, U)$ με $\Gamma(v)$ συμβολίζουμε το σύνολο των κορυφών u που είναι άκρα τόξων με αρχή την κορυφή v (**γείτονες της** v), δηλαδή

$$\Gamma(v) = \{u \in X : (v, u) \in U\}$$

Παράδειγμα: Για το γράφημα τόξων $G = (V, U)$ όπου $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ και

$$U = \{(v_1, v_3), (v_1, v_6), (v_2, v_5), (v_3, v_2), (v_3, v_4), (v_5, v_4), (v_6, v_3), (v_6, v_5), (v_7, v_1), (v_7, v_2)\}.$$

Έχουμε ότι



$$\begin{aligned}
\Gamma(v_1) &= \{v_3, v_6\}\\
\Gamma(v_2) &= \{v_5\}\\
\Gamma(v_3) &= \{v_2, v_4\}\\
\Gamma(v_4) &= \emptyset\\
\Gamma(v_5) &= \{v_4\}\\
\Gamma(v_6) &= \{v_3, v_5\}\\
\Gamma(v_7) &= \{v_1, v_2\}.
\end{aligned}$$

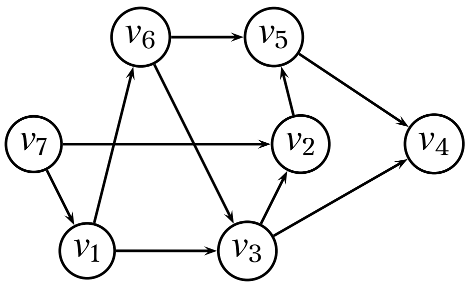
Σε κάθε γράφημα τόξων $G = (V, U)$ **χωρίς κυκλώματα** αντιστοιχεί μια (μοναδική) συνάρτηση $g : V \to \mathbb{N}$ η οποία ονομάζεται **συνάρτηση Grundy** - **Sprague** του G.

Συγκεκριμένα, για κάθε κορυφή v του γραφήματος ορίζουμε

$$g(v) = \operatorname{mex}\{g(u) : \text{ για κάθε } u \in \Gamma(v)\}.$$

Η συνάρτηση g τοποθετεί στις κορυφές του γραφήματος G φυσικούς αριθμούς, έτσι ώστε κάθε κορυφή v να έχει τιμή $g(v)$ τον ελάχιστο φυσικό αριθμό που δεν έχει τοποθετηθεί στους γείτονές της. Η τιμή $g(v)$ ονομάζεται g-**value** της κορυφής v.

Παράδειγμα: Να βρεθούν οι τιμές της συνάρτησης g του γραφήματος τόξων G:



$$\begin{aligned}\Gamma(v_1) &= \{v_3, v_6\}\\ \Gamma(v_2) &= \{v_5\}\\ \Gamma(v_3) &= \{v_2, v_4\}\\ \Gamma(v_4) &= \emptyset\\ \Gamma(v_5) &= \{v_4\}\\ \Gamma(v_6) &= \{v_3, v_5\}\\ \Gamma(v_7) &= \{v_1, v_2\}.\end{aligned}$$

Το G δεν έχει κυκλώματα, άρα ορίζεται η συνάρτηση Grundy - Sprague σ' αυτό. Στόχος μας είναι να συμπληρώσουμε τις κενές θέσεις στον επόμενο πίνακα

v	v_1	v_2	v_3	v_4	v_5	v_6	v_7
$g(v)$							

$$g(v) = \text{mex}\{g(u) : \text{ για κάθε } u \in \Gamma(v)\},$$

Έχουμε ότι

$$\begin{aligned}\Gamma(v_1) &= \{v_3, v_6\}\\ \Gamma(v_2) &= \{v_5\}\\ \Gamma(v_3) &= \{v_2, v_4\}\\ \Gamma(v_4) &= \emptyset\\ \Gamma(v_5) &= \{v_4\}\\ \Gamma(v_6) &= \{v_3, v_5\}\\ \Gamma(v_7) &= \{v_1, v_2\}.\end{aligned}$$

οπότε

$$\begin{aligned}g(v_1) &= \text{mex}\{\text{g}(\text{v}_3), \text{g}(\text{v}_6)\}\\ g(v_2) &= \text{mex}\{\text{g}(\text{v}_5)\}\\ g(v_3) &= \text{mex}\{\text{g}(\text{v}_2), \text{g}(\text{v}_4)\}\\ g(v_4) &= \text{mex}\emptyset\\ g(v_5) &= \text{mex}\{\text{g}(\text{v}_4)\}\\ g(v_6) &= \text{mex}\{\text{g}(\text{v}_3), \text{g}(\text{v}_5)\}\\ g(v_7) &= \text{mex}\{\text{g}(\text{v}_1), \text{g}(\text{v}_2)\}.\end{aligned}$$

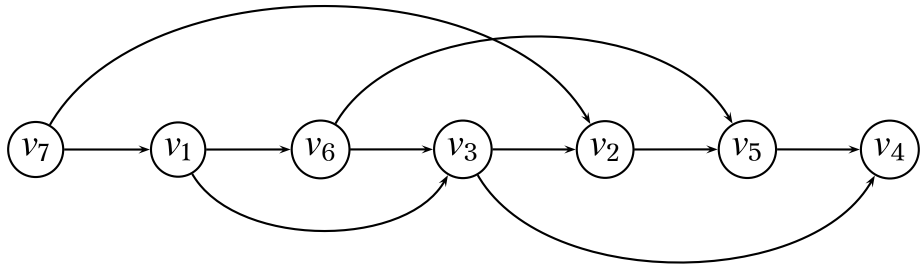
Παρατηρούμε ότι:
Για να υπολογισθεί το $g(v_1)$ πρέπει να υπολογισθούν πρώτα τα $g(v_3)$, $g(v_6)$.
Για να υπολογισθεί το $g(v_2)$ πρέπει να υπολογισθεί πρώτα το $g(v_5)$,
Για να υπολογισθεί το $g(v_3)$ πρέπει να υπολογισθούν πρώτα τα $g(v_2)$, $g(v_4)$.
Για να υπολογισθεί το $g(v_5)$ πρέπει να υπολογισθεί πρώτα το $g(v_4)$,
Για να υπολογισθεί το $g(v_6)$ πρέπει να υπολογισθούν πρώτα τα $g(v_3)$, $g(v_5)$.
Για να υπολογισθεί το $g(v_7)$ πρέπει να υπολογισθούν πρώτα τα $g(v_1)$, $g(v_2)$.
Αντίθετα, για τον υπολογισμό του $g(v_4)$ δεν χρειάζονται άλλες τιμές της g και ισχύει ότι $g(v_4) = \text{mex}\emptyset = 0$.

Επομένως, για να υπολογίσουμε τις τιμές της συνάρτησης g πρέπει να βρούμε (αν υπάρχει) μια σειρά στις κορυφές του, σύμφωνα με την οποία να υπολογίζουμε την τιμή $g(v)$ μια κορυφής αφού πρώτα έχουμε ήδη υπολογίσει τις τιμές $g(u)$ όλων των γειτόνων u της v.

Συνεπώς, ο υπολογισμός της συνάρτησης g είναι πρόβλημα προτεραιοτήτων, επομένως μπορεί να λυθεί υπολογίζοντας μια τοπολογική διάταξη στο σύνολο των κορυφών του γραφήματος G. Η διαφοροποίηση στο πρόβλημα αυτό είναι ότι το τόξο (v, u) λειτουργεί ανάποδα, αφού για τον υπολογισμό της $g(v)$ χρειαζόμαστε την τιμή $g(u)$.

Άρα, για το πρόβλημα αυτό, θα βρούμε μια τοπολογική διάταξη στο γράφημα G και έπειτα θα υπολογίσουμε την συνάρτηση με την ανάστροφη σειρά.

Μια τέτοια διάταξη απεικονίζεται στο επόμενο σχήμα



(από την κορυφή v_i ξεκινάει βέλος προς την κορυφή v_j αν ο υπολογισμός της $g(v_i)$ απαιτεί τον υπολογισμό της $g(v_j)$).

Θα υπολογίσουμε τις τιμές της συνάρτησης g με την ανάστροφη σειρά

$$v_4, v_5, v_2, v_3, v_6, v_1, v_7$$

οπότε σε κάθε βήμα οι τιμές της g που απαιτούνται έχουν υπολογισθεί σε κάποιο προηγούμενο βήμα.

Πράγματι, έχουμε τα εξής:

1) $g(v_4) = \mathrm{mex}\emptyset = 0.$

v	v_1	v_2	v_3	v_4	v_5	v_6	v_7
$g(v)$				0			

2) $g(v_5) = \mathrm{mex}\{g(v_4)\} = \mathrm{mex}\{0\} = 1.$

v	v_1	v_2	v_3	v_4	v_5	v_6	v_7
$g(v)$				0	1		

3) $g(v_2) = \mathrm{mex}\{g(v_5)\} = \mathrm{mex}\{1\} = 0.$

v	v_1	v_2	v_3	v_4	v_5	v_6	v_7
$g(v)$		0		0	1		

4) $g(v_3) = \mathrm{mex}\{g(v_2), g(v_4)\} = \mathrm{mex}\{0, 0\} = \mathrm{mex}\{0\} = 1.$

v	v_1	v_2	v_3	v_4	v_5	v_6	v_7
$g(v)$		0	1	0	1		

5) $g(v_6) = \mathrm{mex}\{g(v_3), g(v_5)\} = \mathrm{mex}\{1, 1\} = \mathrm{mex}\{1\} = 0.$

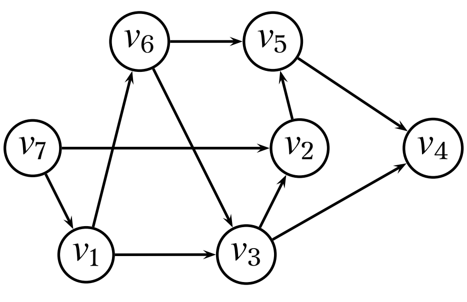
v	v_1	v_2	v_3	v_4	v_5	v_6	v_7
$g(v)$		0	1	0	1	0	

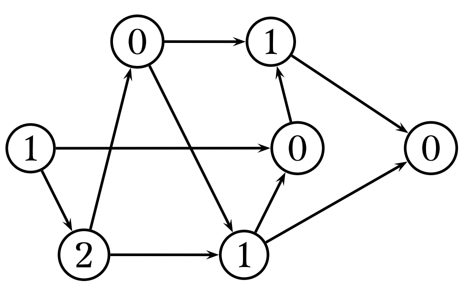
6) $g(v_1) = \mathrm{mex}\{g(v_3), g(v_6)\} = \mathrm{mex}\{1, 0\} = 2.$

v	v_1	v_2	v_3	v_4	v_5	v_6	v_7
$g(v)$	2	0	1	0	1	0	

7) Τέλος, $g(v_7) = \mathrm{mex}\{g(v_1), g(v_2)\} = \mathrm{mex}\{2, 0\} = 1.$

v	v_1	v_2	v_3	v_4	v_5	v_6	v_7
$g(v)$	2	0	1	0	1	0	1.





Παρατήρηση: Η συνάρτηση Grundy - Sprague ικανοποιεί τις παρακάτω ιδιότητες

- Αν για την κορυφή v ισχύει ότι $g(v) \neq 0$, τότε υπάρχει $u \in \Gamma(v)$ με $g(u) = 0$.
- Αν για την κορυφή v ισχύει ότι $g(v) = 0$, τότε για κάθε $u \in \Gamma(v)$ ισχύει ότι $g(u) \neq 0$.

Μια εφαρμογή της συνάρτησης Grundy - Sprague είναι ότι

- αποδεικνύει ότι υπάρχει στρατηγική νίκης σε συνδυαστικά παιχνίδια δύο παιχτών όπου έχουμε αποκλείσει την ισοπαλία (π.χ. σκάκι), δηλαδή αποδεικνύει ότι κάποιος από τους δύο παίχτες μπορεί πάντα να κερδίζει (αρκεί να γνωρίζει την στρατηγική).
- δίνει την στρατηγική νίκης στο παιχνίδι αυτό, αρκεί να είναι υπολογιστικά εφικτός ο υπολογισμός της συνάρτησης g. Η στρατηγική νίκης είναι (πάντα) να φτάνουμε σε καταστάσεις με τιμή g ίση με 0.

Άσκηση 13 (Μια παραλλαγή του Nim).
Δύο παίχτες A και B παίξουν το εξής παιχνίδι:

- *Υπάρχουν δύο σωροί με σπίρτα.*
  Ο πρώτος σωρός αποτελείται από x σπίρτα.
  Ο δεύτερος σωρός αποτελείται από y σπίρτα.
- *Οι παίχτες παίξουν εναλλάξ κάνοντας* **μια κίνηση ο καθένας***: Κάθε παίχτης αφαιρεί ένα ή περισσότερα σπίρτα*
  - *ή από τον πρώτο σωρό,*
  - *ή από τον δεύτερο σωρό,*
  - *ή τον ίδιο αριθμό σπίρτων και από τους δύο σωρούς.*
- **Χάνει** *ο παίχτης που δεν μπορεί να κάνει κίνηση.*
- *Για παράδειγμα, αν ο πρώτος σωρός περιέχει 12 σπίρτα και ο δεύτερος σωρός 8 σπίρτα ποιος από τους δύο παίχτες θα κερδίσει;*

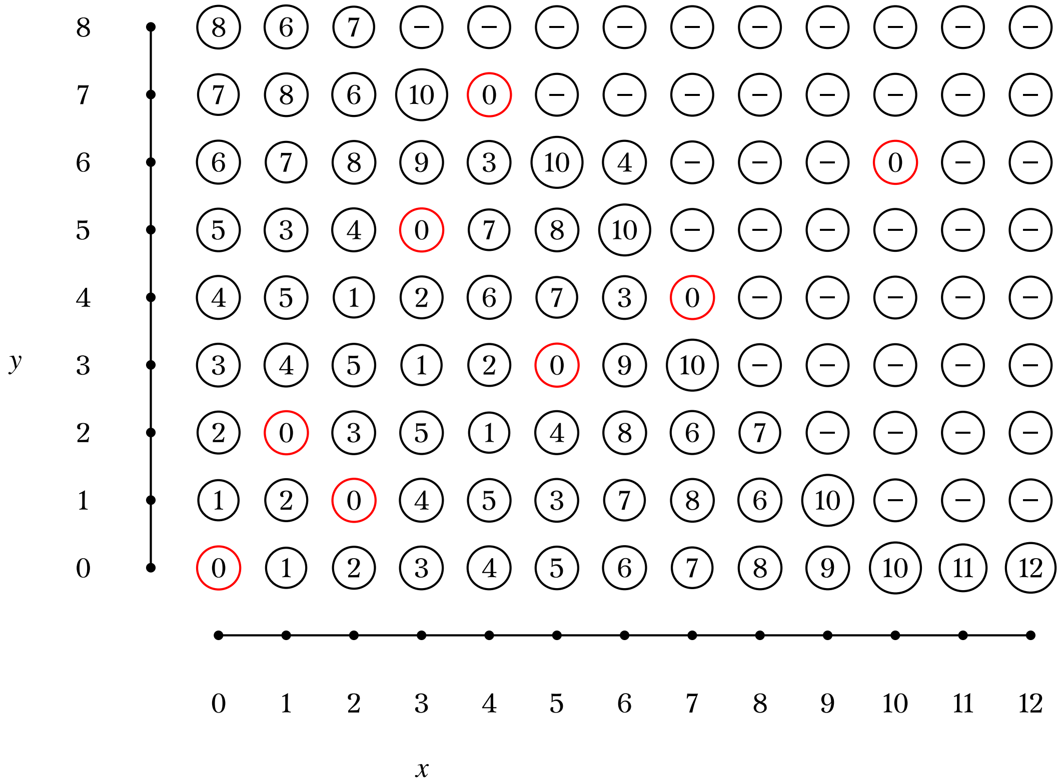
Λύση. Μπορούμε να μοντελοποιήσουμε το παιχνίδι ως μια ακολουθία καταστάσεων (ένα μονοπάτι κορυφών πάνω ένα γράφημα τόξων).

- Οι κορυφές του γραφήματος είναι οι καταστάσεις του παιχνιδιού: Κάθε κορυφή αντιστοιχεί σε ένα διατεταγμένο ζεύγος (x, y) όπου x ο αριθμός των σπίρτων στον πρώτο σωρό και y ο αριθμός των σπίρτων στον δεύτερο σωρό.

  Ισχύει ότι $0 \leq x \leq 12$ και $0 \leq y \leq 8$. Συνολικά υπάρχουν $13 \times 9 = 117$ καταστάσεις (κορυφές).
- Αρχικά το παιχνίδι βρίσκεται στην κατάσταση (κορυφή) $(12, 8)$. Με κάθε κίνηση των παιχτών το παιχνίδι αλλάζει καταστάσεις. Οι επιτρεπτές κινήσεις αντιστοιχούν στην μετακίνηση οριζόντια (προς τα αριστερά) ή κατακόρυφα (προς τα κάτω) ή διαγώνια (προς το κέντρο) οσαδήποτε βήματα. Κερδίζει ο παίχτης που θα φτάσει πρώτος στην κορυφή $(0, 0)$.

- Στο επόμενο σχήμα απεικονίζονται μόνο οι κορυφές του γραφήματος. Τα τόξα ορίζονται από τις επιτρεπτές κινήσεις.

  Επίσης, έχουν σημειωθεί οι τιμές της συνάρτησης Grundy - Sprague. Οι μηδενικές τιμές (τιμές με κόκκινο) ορίζουν την νικηφόρα στρατηγική του παιχνιδιού.



- Στο παράδειγμα, μπορεί να κερδίζει πάντα ο πρώτος παίχτης, διότι αφαιρώντας 2 σπίρτα και από τους δύο σωρούς καταλήγει στην κατάσταση $(10, 6)$ η οποία έχει τιμή g ίση με 0.

□

Παρατήρηση: Σε κάθε συνδυαστικό παιχνίδι δύο παιχτών (χωρίς ισοπαλία) μπορούμε να κατασκευάσουμε το γράφημα των καταστάσεων του παιχνιδιού. Το γράφημα αυτό δεν θα περιέχει κυκλώματα (αφού δεν επιτρέπεται ισοπαλία).

Πόρισμα 56. *Κάθε συνδυαστικό παιχνίδι δύο παιχτών (χωρίς ισοπαλία) έχει συνάρτηση g. Επομένως, μπορεί να κερδίζει πάντα ή ο πρώτος, ή ο δεύτερος παίχτης. (Αρκεί να γνωρίζει τις καταστάσεις που μηδενίζουν την συνάρτηση g.)*

34. ΔΙΑΤΑΞΗ ΠΑΡΑΓΩΓΗΣ - ΣΤΑΘΜΕΣ

Ένας εναλλακτικός τρόπος υπολογισμού της τοπολογικής διάταξης είναι χρησιμοποιώντας την μέθοδο Demoucron:

Σε κάθε γράφημα τόξων με p κορυφές σχηματίζουμε ένα πίνακα (με p γραμμές) ως εξής :

Στις πρώτες p στήλες $v_1, v_2, \ldots, v_p$ τοποθετούμε 0 και 1, όπως ακριβώς στη μήτρα γειτονικότητας του γραφήματος, (συνήθως παραλείπουμε τα 0).

Τις επόμενες στήλες $S_0, S_1, \ldots,$ τις συμπληρώνουμε διαδοχικά, χρησιμοποιώντας την εξής αναδρομική διαδικασία:

Στη στήλη S_0 γράφουμε στις αντίστοιχες γραμμές το άθροισμα των 1 κάθε γραμμής (δηλαδή, τους βαθμούς εξόδου των κόμβων $v_1, v_2, \ldots, v_p$). Γράφουμε κάτω από τον πίνακα τους κόμβους με βαθμό εξόδου 0. Οι κόμβοι αυτοί θα τοποθετηθούν στην τελευταία στάθμη. (Δεν θα ασχοληθούμε άλλο με τις γραμμές που αντιστοιχούν στους κόμβους αυτούς. Γράφουμε $\times$ σε κάθε στήλη, δεξιά από κάθε 0).

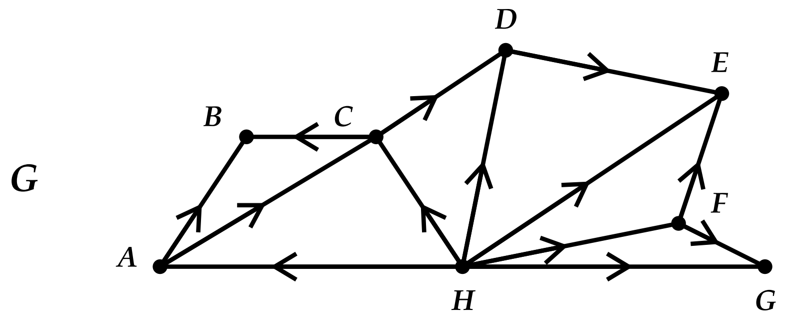
Έστω τώρα, ότι έχουμε συμπληρώσει μέχρι και τη στήλη S_n, $(n \geq 0)$ και έχουμε γράψει κάτω από τον πίνακα και τους κόμβους $v_i, \ldots, v_j$ που αντιστοιχούν στα 0 της στήλης S_n, (οι οποίοι θα είναι οι κόμβοι στην n-στή πριν από το τέλος στάθμη).

Στη στήλη S_{n+1} γράφουμε τα στοιχεία της στήλης S_n, καθένα μειωμένο κατά τόσες μονάδες, όσα είναι τα 1 που περιέχονται στις στήλες $v_i, \ldots, v_j$ της αντίστοιχης γραμμής. Οι κόμβοι που αντιστοιχούν στα 0 της στήλης αυτής, είναι τα στοιχεία της $(n+1)$-στής πριν από το τέλος στάθμης.

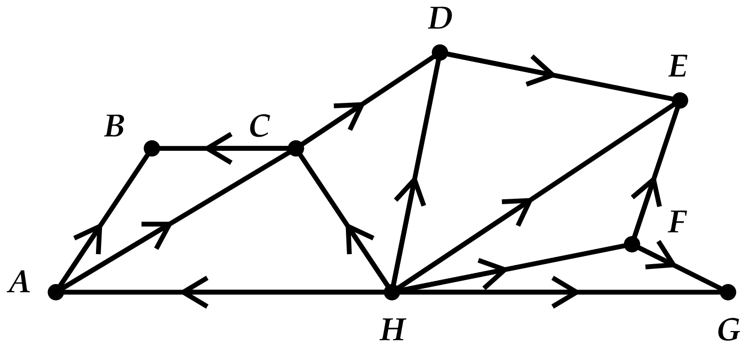
Η διαδικασία ολοκληρώνεται όταν σχηματισθεί μια στήλη S_l που αποτελείται μόνο από 0 και $\times$.

Παράδειγμα : Έστω A, B, C, D, E, F, G, H οι τεχνολογικές διαδικασίες, που υπόκεινται σε μια σχέση "τεχνολογικής προτεραιότητας" ως εξής : $A < B$, $A < C$, $C < B$, $C < D$, $D < E$, $F < E$, $F < G$, $H < A$, $H < C$, $H < D$, $H < E$, $H < F$, $H < G$, (γράφουμε $x < y$, όταν η δραστηριότητα x προηγείται της y).

Σχηματίζουμε το αντίστοιχο γράφημα τόξων G :



G

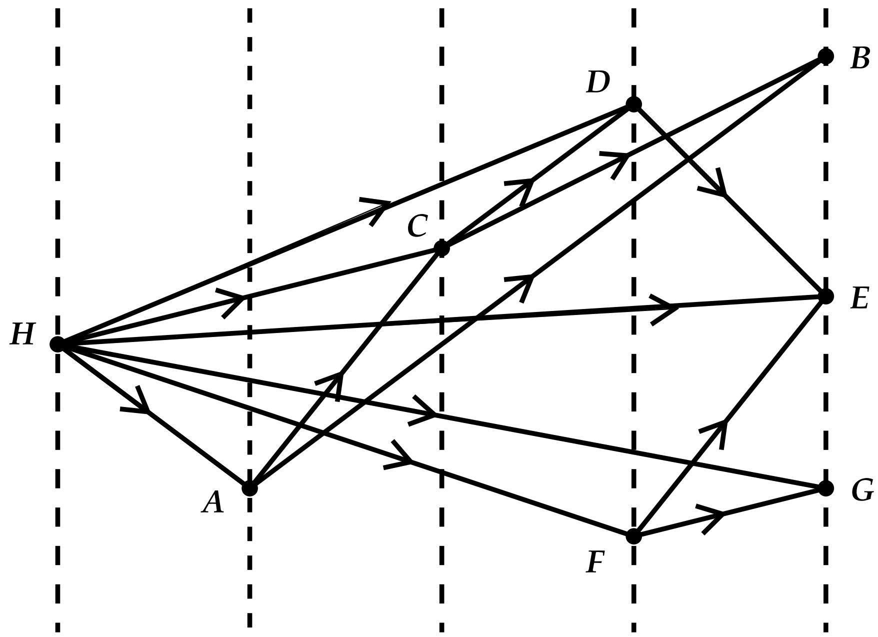


Η παραπάνω διαδικασία δίνει :

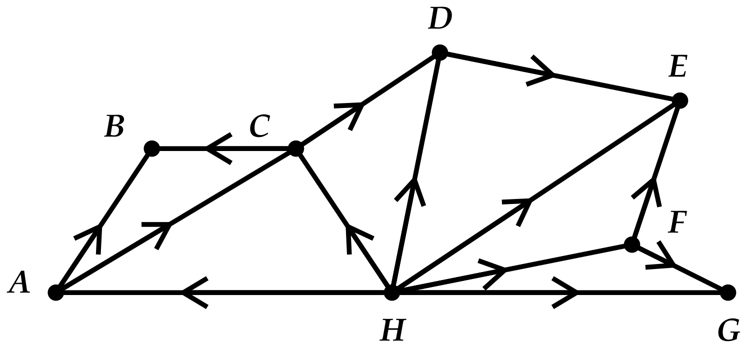
	A	B	C	D	E	F	G	H	S_0	S_1	S_2	S_3	S_4
A													
B													
C													
D													
E													
F													
G													
H													

Τα B, E, G βρίσκονται στην τελευταία στάθμη.
Τα D, F βρίσκονται στην προτελευταία στάθμη.
Το C βρίσκεται στην τρίτη από το τέλος στάθμη.
Το A βρίσκεται στην τετάρτη από το τέλος στάθμη.
Το H βρίσκεται στην πρώτη στάθμη,

οπότε



G

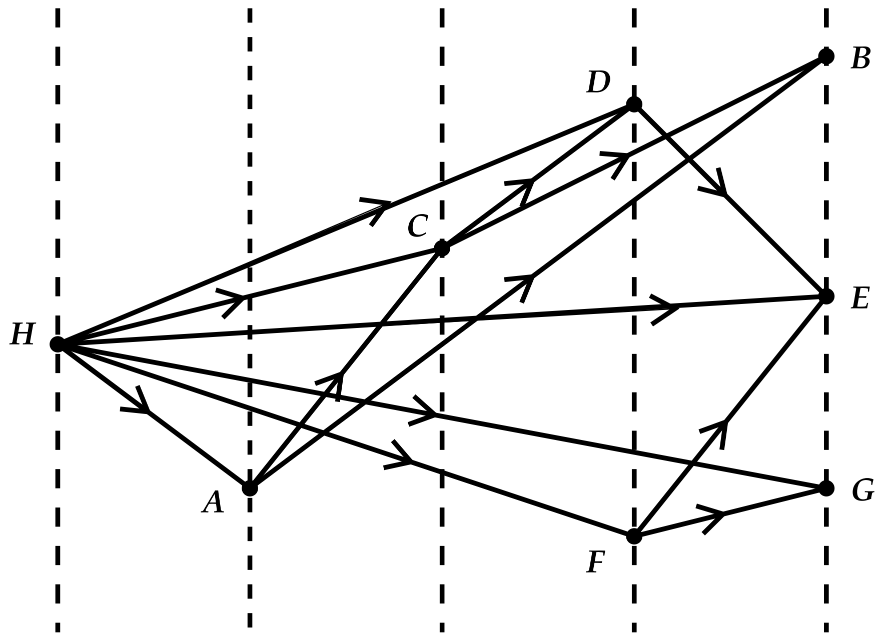


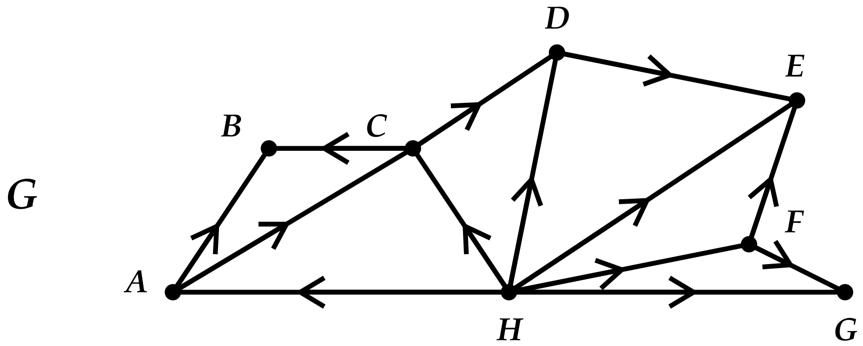
Η παραπάνω διαδικασία δίνει :

	A	B	C	D	E	F	G	H	S_0	S_1	S_2	S_3	S_4
A		1	1										
B													
C		1		1									
D					1								
E													
F					1		1						
G													
H	1		1	1	1	1	1						

Τα B, E, G βρίσκονται στην τελευταία στάθμη.
Τα D, F βρίσκονται στην προτελευταία στάθμη.
Το C βρίσκεται στην τρίτη από το τέλος στάθμη.
Το A βρίσκεται στην τετάρτη από το τέλος στάθμη.
Το H βρίσκεται στην πρώτη στάθμη,

οπότε



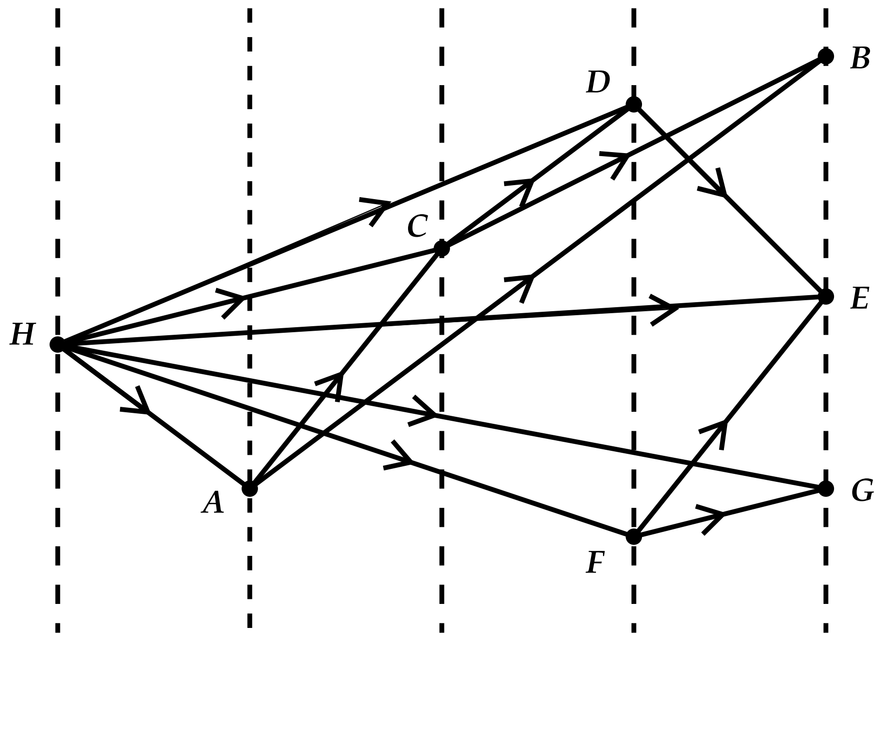


Η παραπάνω διαδικασία δίνει :

	A	B	C	D	E	F	G	H	S_0	S_1	S_2	S_3	S_4
A		1	1						2	1	1	0	$\times$
B									0	$\times$	$\times$	$\times$	$\times$
C		1		1					2	1	0	$\times$	$\times$
D					1				1	0	$\times$	$\times$	$\times$
E									0	$\times$	$\times$	$\times$	$\times$
F					1		1		2	0	$\times$	$\times$	$\times$
G									0	$\times$	$\times$	$\times$	$\times$
H	1		1	1	1	1	1		6	4	2	1	0

Τα B, E, G βρίσκονται στην τελευταία στάθμη.
Τα D, F βρίσκονται στην προτελευταία στάθμη.
Το C βρίσκεται στην τρίτη από το τέλος στάθμη.
Το A βρίσκεται στην τετάρτη από το τέλος στάθμη.
Το H βρίσκεται στην πρώτη στάθμη,

οπότε



35. ΕΦΑΡΜΟΓΗ ΣΤΟ ΧΡΟΝΙΚΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ (ΕΝΑ ΠΑΡΑΔΕΙΓΜΑ)

Υποθέτουμε ότι για τις εργασίες ενός έργου ισχύει ο παρακάτω πίνακας, όπου το i (αντίστοιχα το j) συμβολίζει την έναρξη (αντίστοιχα τη λήξη) μιας συγκεκριμένης δραστηριότητας σε ένα έργο, ενώ ο χρόνος t_{ij} είναι ο αναμενόμενος χρόνος για την πραγματοποίηση της δραστηριότητας (i, j) του έργου.

Δραστηριότητες		Χρόνος
i	j	t_{ij}
1	2	5
1	3	6
1	4	6
2	3	3
2	5	4
2	6	3
3	4	5
3	6	6
3	8	4
3	9	7
4	7	7
5	9	2
5	11	4
6	7	4
6	9	5
7	8	7
7	11	2
8	9	3
8	10	2
9	10	3
10	12	8
11	12	9

Θεωρούμε ότι η έναρξη της δραστηριότητας (j, k) λαμβάνει χώρα μόνο όταν έχουν πραγματοποιηθεί όλες οι δραστηριότητες (i, j), (για $j \neq 1$).

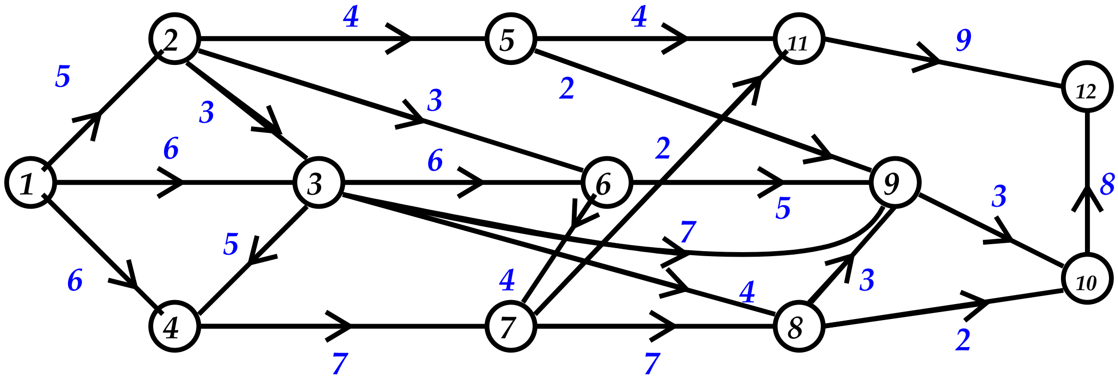
Από τα παραπάνω προκύπτει ένα γράφημα τόξων με κόμβους τις ενάρξεις και τις λήξεις των δραστηριοτήτων, τόξα τις αντίστοιχες δραστηριότητες και αριθμούς στα τόξα οι οποίοι δίνουν τους αναμενόμενους χρόνους για κάθε τέτοια δραστηριότητα. Ο κόμβος 1 δίνει την έναρξη και ο κόμβος 12 τη λήξη του έργου.

Το γράφημα δεν πρέπει να περιέχει κυκλώματα.

Ζητάμε τον ενωρίτερο χρόνο κατά τον οποίο μπορεί να ολοκληρωθεί το έργο. Αρκεί λοιπόν να βρούμε διαδοχικά τον ενωρίτερο χρόνο ολοκλήρωσης για να φτάσουμε σε κάθε κόμβο, υποθέτοντας ότι ο ενωρίτερος χρόνος για το 1 (έναρξη) είναι 0.

Ο ζητούμενος ενωρίτερος χρόνος ολοκλήρωσης του έργου, είναι λοιπόν προφανώς ο ενωρίτερος χρόνος για τον κόμβο 12 (λήξη του έργου).

Οι διαδοχικοί χρόνοι είναι σημειωμένοι δίπλα στην κάθε κορυφή του γραφήματος.



Βλέπουμε λοιπόν ότι ο ενωρίτερος χρόνος για την ολοκλήρωση του έργου με τα δεδομένα του προηγούμενου πίνακα είναι 41.

Ο δρόμος (1, 2, 3, 4, 7, 8, 9, 10, 12) που αντιστοιχεί στον ανωτέρω χρόνο ονομάζεται **κρίσιμος δρόμος** του έργου και οι αντίστοιχες δραστηριότητες **κρίσιμες δραστηριότητες**.

Άσκηση 14. *Πως θα αλλάξει ο χρόνος ολοκλήρωσης ενός έργου*

(1) αν όλοι οι χρόνοι των εργασιών πολλαπλασιαστούν επί μια σταθερά c;

(2) αν όλοι οι χρόνοι των εργασιών αυξηθούν κατά μια σταθερά c;

Ασκήσεις προς επίλυση

(1) Έστω $A, B, \Gamma, \Delta, E, Z$ οι τεχνολογικές διαδικασίες μιας παραγωγής, που υπόκεινται στη σχέση τεχνολογικής προτεραιότητας ως εξής:

$$A < \Delta,\ \ B < A,\ \ B < \Delta,\ \ B < E,\ \ B < Z,$$

$$\Gamma < B,\ \ \Gamma < E,\ \ E < \Delta,\ \ Z < A,\ \ Z < E$$

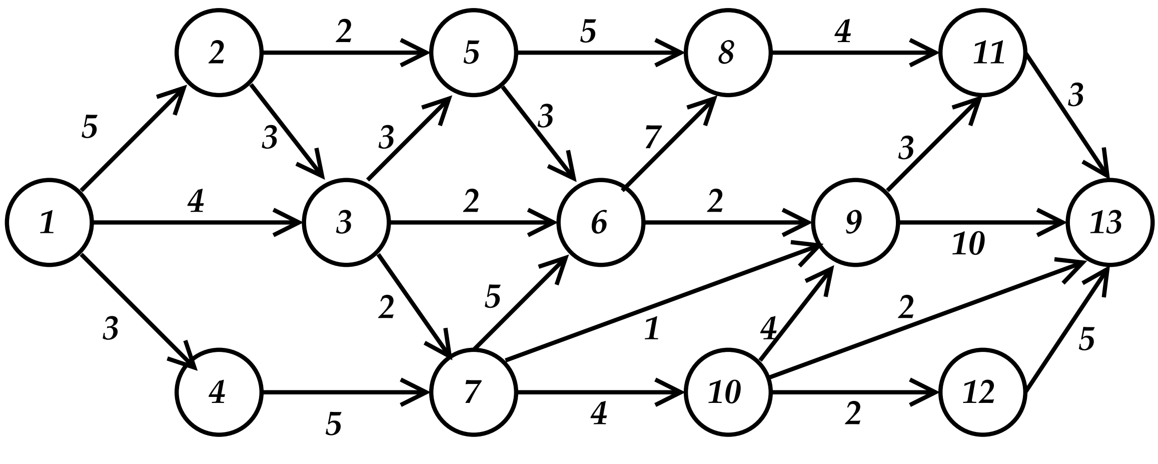
α) Να σχεδιαστεί το σχετικό γράφημα G.

β) Με εφαρμογή της μεθόδου Demoucron να ευρεθούν οι στάθμες των κορυφών του G.

γ) Να δοθεί γραφικά το σχετικό αποτέλεσμα, που περιγράφει την διάταξη παραγωγής.

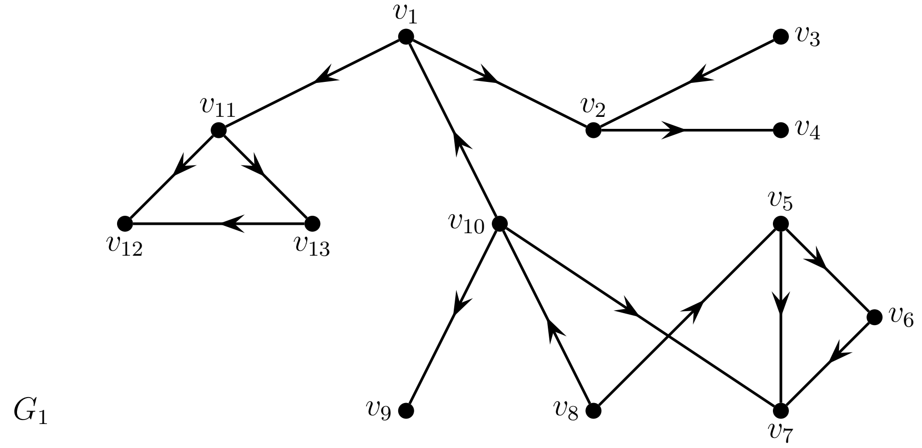
	A	B	Γ	Δ	E	Z						
A												
B												
Γ												
Δ												
E												
Z												

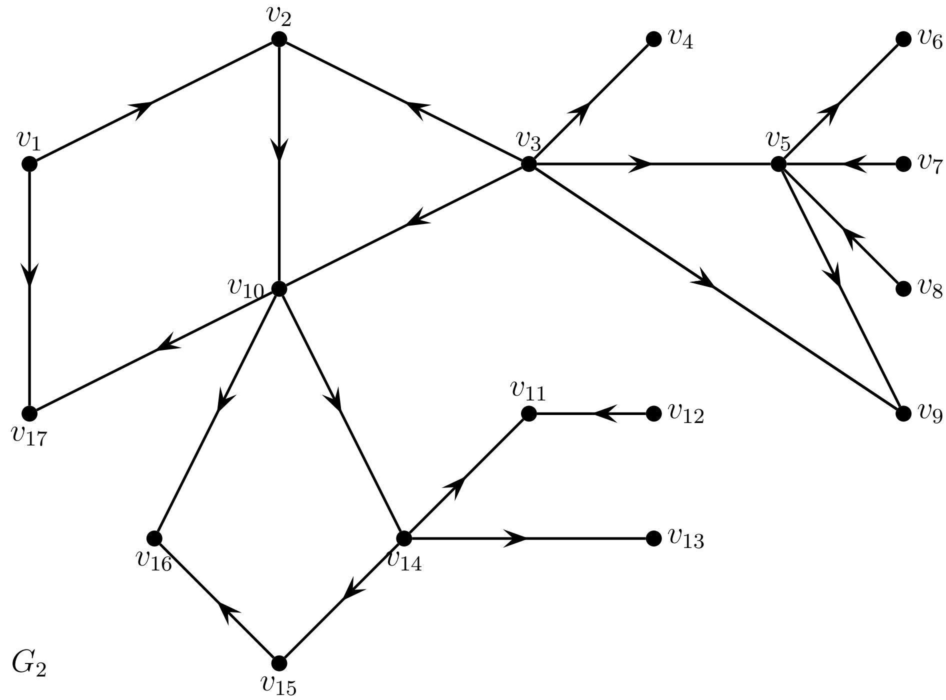
(2) Να βρεθεί ο ενωρίτερος χρόνος ολοκλήρωσης και ο κρίσιμος δρόμος του έργου που αρχίζει από τον κόμβο 1 και τελειώνει στον κόμβο 13, αν το έργο αυτό περιγράφεται από το παρακάτω γράφημα τόξων.



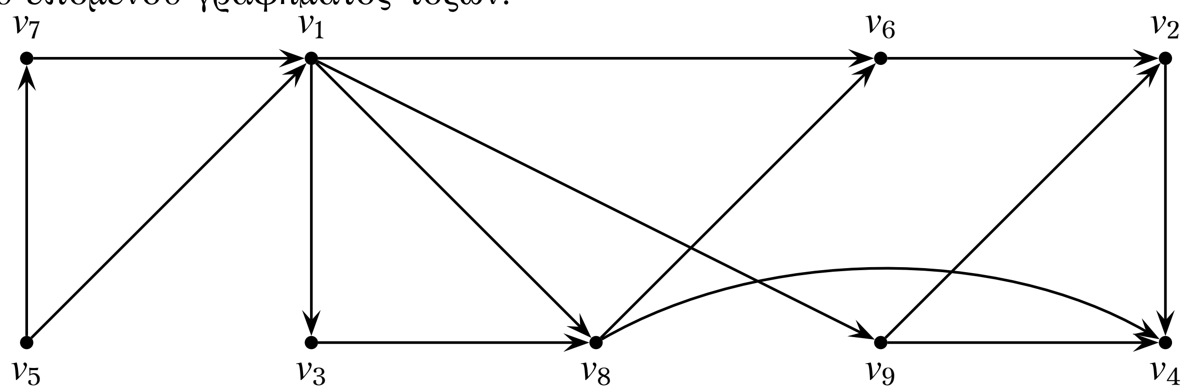
(Ο χρόνος για κάθε δραστηριότητα (i, j) σημειώνεται πάνω στο αντίστοιχο τόξο. Θεωρούμε ότι, για κάθε $j \neq 1$, η έναρξη της δραστηριότητας (j, k) λαμβάνει χώρα μόνο όταν έχουν πραγματοποιηθεί όλες οι δραστηριότητες (i, j)).

(3) Να βρεθεί η τοπολογική διάταξη για τα παρακάτω γραφήματα:





(4) Να υπολογισθούν οι τιμές της συνάρτησης Grundy - Sprague για τις κορυφές του επόμενου γραφήματος τόξων.



(5) Πάνω σε ένα τραπέζι βρίσκονται 25 σπίρτα. Δύο παίχτες κάνουν εναλλάξ την εξής κίνηση: Αφαιρούν 1, 2 ή 3 σπίρτα από το τραπέζι. Ο παίχτης που θα πάρει το τελευταίο σπίρτο χάνει. Να δειχθεί ότι ο δεύτερος παίχτης μπορεί πάντα να κερδίζει το παιχνίδι. (Δοκιμάστε να παίξετε το παιχνίδι με 5, 9 και 13 σπίρτα)

36. ΔΥΝΑΜΙΚΑ ΜΟΝΤΕΛΑ ΓΡΑΦΗΜΑΤΩΝ

Στην ενότητα αυτή παρουσιάζονται ορισμένα παραδείγματα που αφορούν γραφήματα τα οποία αλλάζουν δυναμικά (π.χ. προστίθενται ή/και διαγράφονται κορυφές ή/και δεσμοί, αλλάζουν οι ετικέτες των κορυφών ή/και των δεσμών).

36.1. Δίκτυα επιρροής. Τα δίκτυα επιρροής μοντελοποιούν το πως μια νέα ιδέα ή άποψη υιοθετείται ή απορρίπτεται από ένα δίκτυο ατόμων. Η μοντελοποίηση γίνεται με βάση την παραδοχή ότι η γνώμη που υιοθετεί ένα άτομο επηρεάζεται από την πίεση που του ασκούν οι επαφές του (peer pressure): Συγκεκριμένα, όσο μεγαλύτερο το ποσοστό των επαφών μας που μοιράζονται ή υιοθετούν μια ιδέα, τόσο πιο πιθανό είναι ότι θα την υιοθετήσουμε και εμείς.

Δίκτυο επιρροής είναι ένα γράφημα δεσμών (ή τόξων) $G = (V, E)$ στο οποίο κάθε κορυφή v έχει μια ετικέτα $l(v)$. Συγκεκριμένα, για κάθε κορυφή v είτε $l(v) = 1$ όταν η v αποδέχεται μια άποψη και χαρακτηρίζεται **ενεργή** είτε $l(v) = 0$, όταν η v δεν αποδέχεται την άποψη και χαρακτηρίζεται **μη ενεργή**.

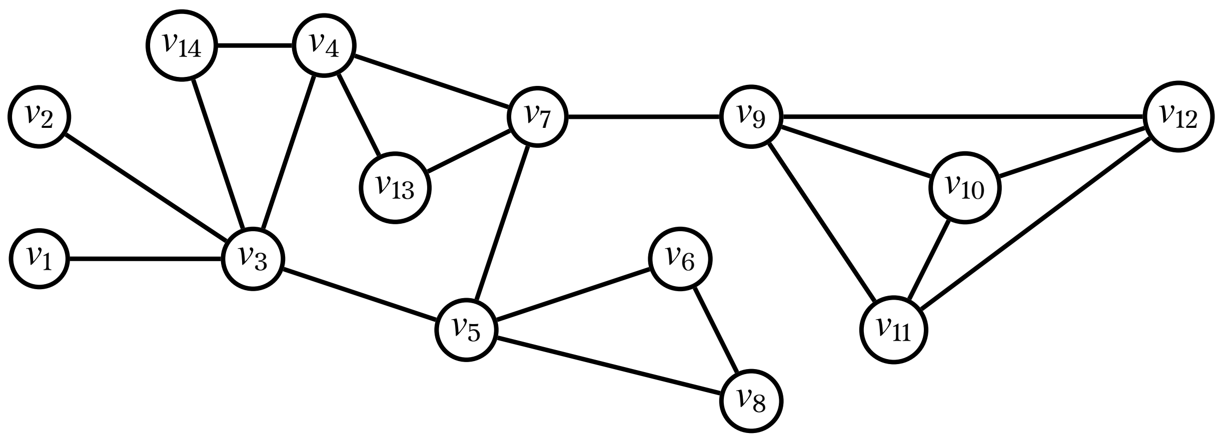
Το μοντέλο λειτουργεί δυναμικά (δηλαδή οι ετικέτες των κορυφών μπορούν να αλλάξουν). Ο βασικός μηχανισμός του μοντέλου περιγράφεται στα επόμενα βήματα:

- Αρχικά όλες οι κορυφές του γραφήματος $G = (V, E)$ είναι μη ενεργές, δηλαδή $l(v) = 0$ για κάθε $v \in V$.

  Επιλέγεται (είτε τυχαία, είτε μέσω κάποιας διαδικασίας) ένα υποσύνολο S των κορυφών του G οι οποίες ενεργοποιούνται, δηλαδή θέτουμε $l(v) = 1$ για κάθε $v \in S$.
- Επαναλαμβάνουμε τα επόμενα βήματα μέχρις ότου δεν υπάρχουν αλλαγές στις ετικέτες των κορυφών.
  - Αν μια κορυφή ειναι ενεργή, παραμένει ενεργή.
  - Αν μια κορυφή είναι μη ενεργή και στην προηγούμενη επανάληψη τουλάχιστον $\theta \cdot 100\%$ από τις γειτονικές[5] της κορυφές είναι ενεργές, τότε και αυτή γίνεται ενεργή (εναλλακτικά, γίνεται ενεργή με κάποια πιθανότητα).

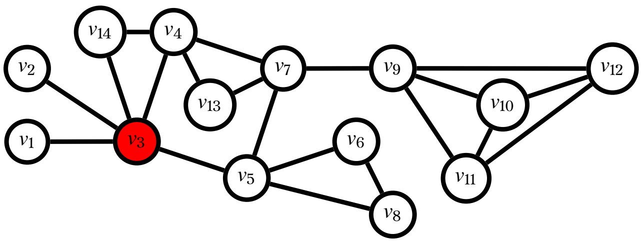
Ο αριθμός $\theta \in (0, 1]$ ονομάζεται **κατώφλι ενεργοποίησης** (activation threshold) και το συγκεκριμένο μοντέλο ονομάζεται **μοντέλο κατωφλίου** (threshold model).

Παράδειγμα:

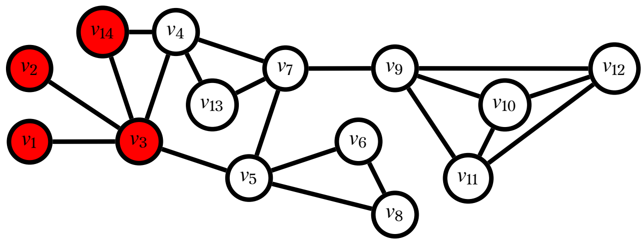


[5] Σε γραφήματα τόξων, στο μοντέλο αυτό θεωρούμε ως γειτονικές τις κορυφές που είναι αρχή τόξων που εισέρχονται στην κορυφή.

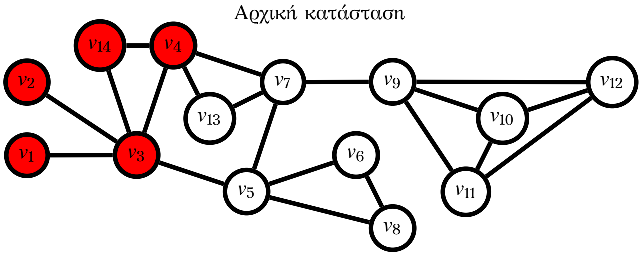
Για παράδειγμα, αν επιλέξουμε $\theta = 50\%$ και ενεργοποιήσουμε την κορυφή v_3 του προηγούμενου γραφήματος, τότε στην πρώτη επανάληψη θα ενεργοποιηθούν οι κορυφές v_1, v_2, v_{14}. Στην επόμενη επανάληψη θα ενεργοποιηθεί η κορυφή v_4. Έπειτα, θα ενεργοποιηθεί η κορυφή v_{13}. Μετά, θα ενεργοποιηθεί η κορυφή v_7. Στην συνέχεια, θα ενεργοποιηθεί η κορυφή v_5. Ακολούθως, θα ενεργοποιηθούν οι κορυφές v_6, v_8. Η διαδικασία θα ολοκληρωθεί εδώ. Οι κορυφές v_9, v_{10}, v_{11}, v_{12} θα παραμείνουν μη ενεργές.



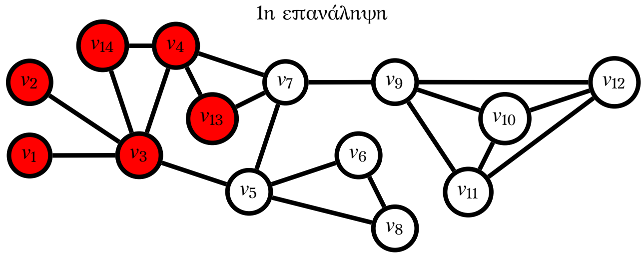
Αρχική κατάσταση



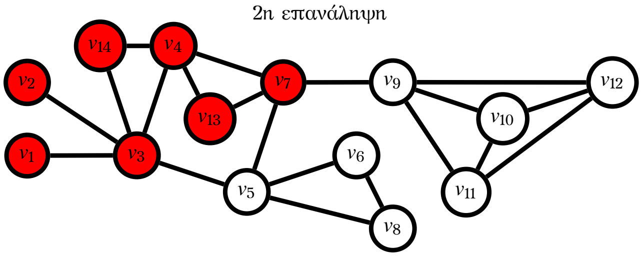
1η επανάληψη



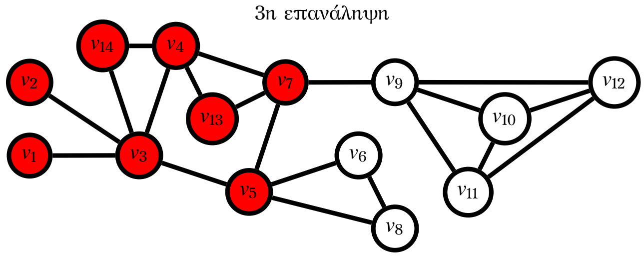
2η επανάληψη



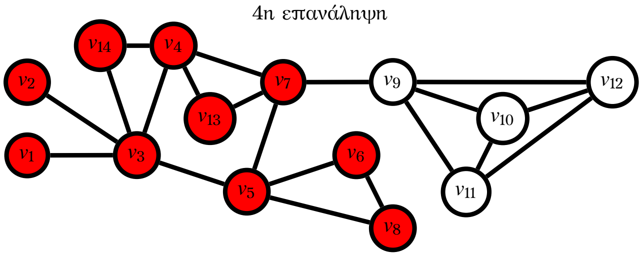
3η επανάληψη



4η επανάληψη



5η επανάληψη



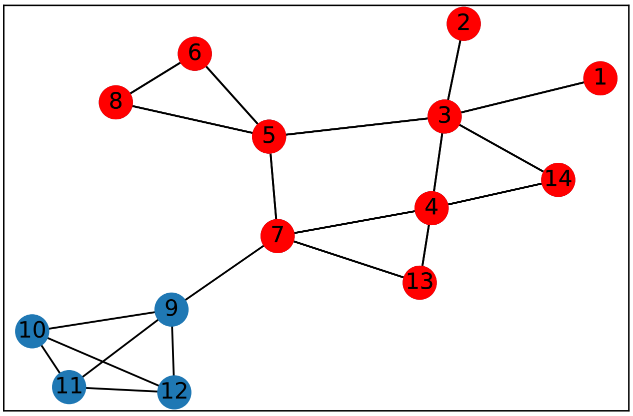
Τελική κατάσταση

Μπορούμε εύκολα να δοκιμάσουμε το μοντέλο αυτό χρησιμοποιώντας την βιβλιοθήκη `networkx`.

```
import networkx as nx
import matplotlib.pyplot as plt

theta = 0.5 #threshold

G = nx.Graph()
n = 14 #A graph with 14 vertices
G.add_nodes_from(range(1,n+1))
E = [[1,3],[2,3],[3,4],[3,5],[3,14],[4,7],[4,13],[4,14],[5,7],[5,6],[5,8],
     [6,8],[7,9],[7,13],[9,10],[9,11],[9,12],[10,11],[10,12],[11,12]]
G.add_edges_from(E)

#Draw the initial graph before running the model
pos = nx.layout.kamada_kawai_layout(G)
```

```
nx.draw_networkx(G,pos)
#create the list of labels setting all labels equal to 0 (label[0] is not used)
#label[i] corresponds to the label of node i
labels = [0 for i in range(1,n+2)]
labels[3] = 1 #set node 3 active

changesMade = True
while(changesMade):
    changesMade = False
    for v in G:
        if(labels[v]==0): #if node v is inactive
            activeneighbors = 0
            for u in G.neighbors(v):
                if(labels[u]==1): activeneighbors+=1
            #if the percentage of active neigbhbors of v
            #is greater than or equal to theta then set v active
            if(activeneighbors/G.degree(v) >= theta):
                changesMade = True
                labels[v] = 1
                print("Message: Node",v,"is now active")

#list of active nodes after running the model
activeNodes = [v for v in G if labels[v]==1]
print("Number of nodes:",len(G),"number of active nodes:",len(activeNodes),
      "percentage of active nodes:",len(activeNodes)/len(G))
nx.draw_networkx(G.subgraph(activeNodes),pos,node_color="red")
plt.show()
```

```
Message: Node 1 is now active
Message: Node 2 is now active
Message: Node 14 is now active
Message: Node 4 is now active
Message: Node 13 is now active
Message: Node 7 is now active
Message: Node 5 is now active
Message: Node 6 is now active
Message: Node 8 is now active
Number of nodes: 14 number of active
   nodes: 10 percentage of active nodes
   0.7142857142857143
```



Υπάρχουν πολλά ερωτήματα που αφορούν αυτό το μοντέλο. Για παράδειγμα, πώς αλλάζει το ποσοστό των ενεργών κορυφών με βάση το κατώφλι ενεργοποίησης και τις αρχικές ενεργές κορυφές; Ποιες κορυφές αρκούν να ενεργοποιηθούν ώστε να πετύχουμε συγκεκριμένο ποσοστό ενεργοποίησης; Σε τι είδους γραφήματα καταλήγουμε σε ομοφωνία (όλοι να έχουν την ίδια γνώμη);

Τι άλλα μοντέλα επιρροής μπορούν να κατασκευασθούν; Για παράδειγμα, είναι γνωστό ότι δεν επηρεαζόμαστε από τις επαφές μας εξίσου, δηλαδή ορισμένες επαφές ασκούν μεγαλύτερη επιρροή. Στην περίπτωση αυτή, πρέπει να χρησιμοποιηθούν βάρη πάνω στις συνδέσεις που εκφράζουν τον βαθμό επιρροής.

36.2. Ανίχνευση κοινοτήτων. Έστω ένα γράφημα που αναπαριστά ένα κοινωνικό δίκτυο. Ένα υποσύνολο κορυφών του ονομάζεται **κοινότητα** (community) όταν τα μέλη του μοιράζονται κάτι κοινό (π.χ. την ίδια άποψη ή τα ιδια ενδιαφέροντα). Στην ενότητα αυτή παρουσιάζεται ένας απλός αλγόριθμος για την ανίχνευση κοινοτήτων σε ένα γράφημα (δεσμών ή τόξων).

Ο αλγόριθμος ονομάζεται **αλγόριθμος διάδοσης ετικετών** (label propagation algorithm) και βασίζεται στην ιδέα ότι οι γειτονικές κορυφές συνήθως ανήκουν στην ίδια κοινότητα, διότι συνήθως τα άτομα προτιμούν να έχουν επαφές με κοινές ιδέες ή ενδιαφέροντα.

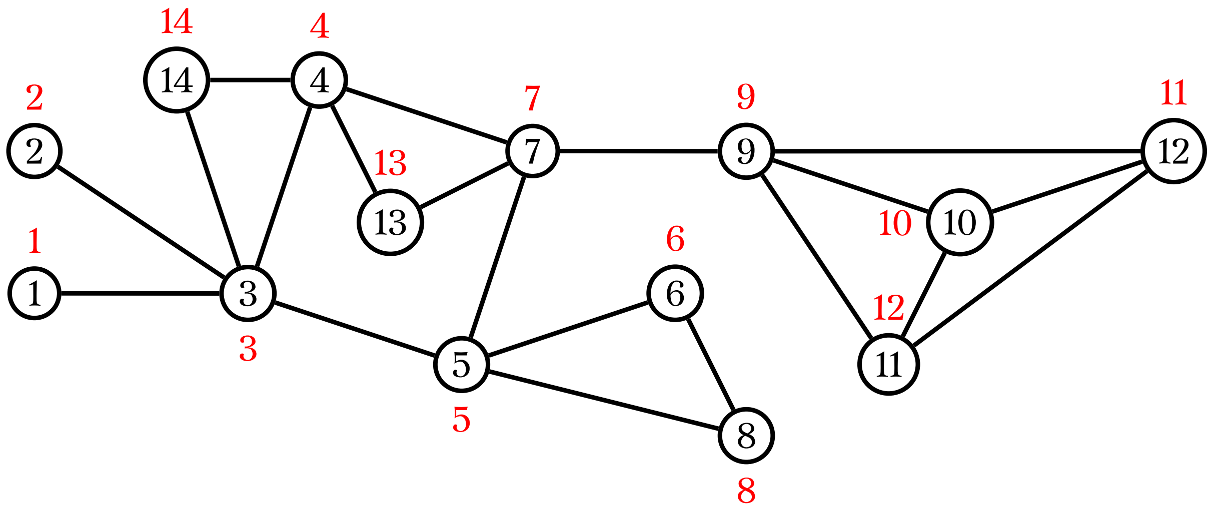
Σε κάθε βήμα του αλγορίθμου εξετάζεται κάθε κορυφή και εντάσσεται στην κοινότητα που ανήκει η πλειοψηφία των γειτόνων της. Η διαδικασία αυτή συγκλίνει σε μια σταθερή διαμέριση των κορυφών του γραφήματος, όπου κάθε κορυφή ανήκει στην ίδια κοινότητα που ανήκει η πλειοψηφία των γειτόνων της.

Συγκεκριμένα, έστω $G = (V, E)$ ένα γράφημα δεσμών.

- Αρχικά κάθε κορυφή v λαμβάνει μια διαφορετική ετικέτα $l(v)$.
- Επαναλαμβάνουμε τα επόμενα βήματα μέχρις ότου δεν υπάρχουν αλλαγές στις ετικέτες των κορυφών.
  - Για κάθε κορυφή v, σε τυχαία σειρά, η κορυφή v λαμβάνει την ετικέτα της πλειοψηφίας των γειτόνων της.
    Αν υπάρχουν πολλές ετικέτες που ισοψηφούν, τότε επιλέγεται τυχαία μια από αυτές.
    Αν η πλειοψηφία έχει την ίδια ετικέτα με την v δεν γίνεται καμιά αλλαγή.

Κατά την διάρκεια εκτέλεσης του αλγορίθμου οι ετικέτες των κοινοτήτων διαδίδονται μέσα στο δίκτυο: οι περισσότερες εξαφανίζονται ενώ άλλες κυριαρχούν. Επειδή η διαμέριση του δικτύου αλλάζει σε κάθε εκτέλεση της επανάληψης, απαιτούνται αρκετές επαναλήψεις μέχρι να συγκλίνει σε μια σταθερή κατάσταση. Παρόλα, αυτά συνήθως όμως ο αλγόριθμος συγκλίνει μετά από ένα μικρό αριθμό επαναλήψεων, που δεν εξαρτάται από το μέγεθος του γραφήματος.

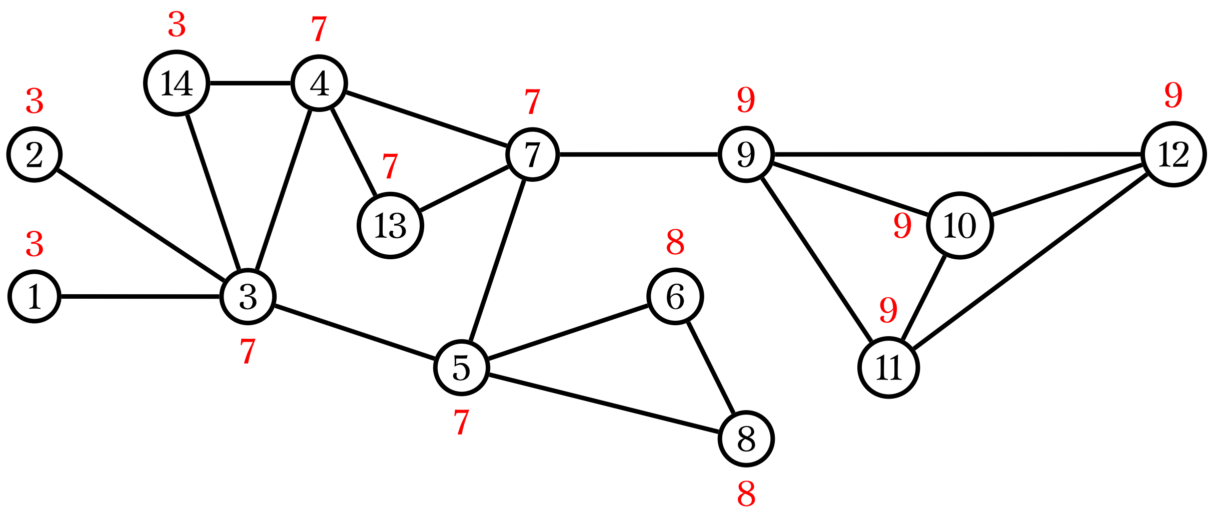
Παράδειγμα:



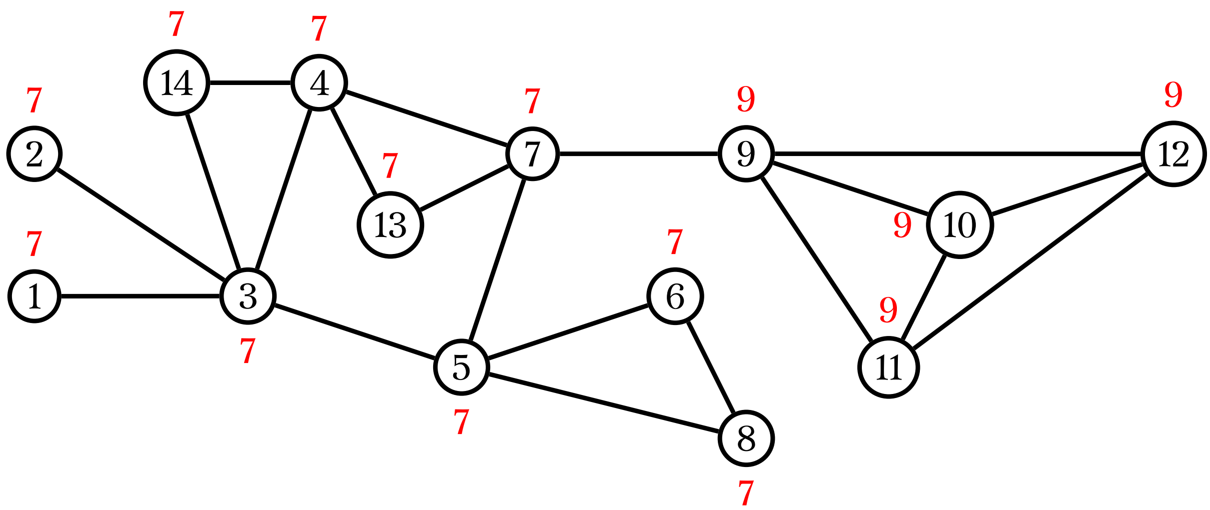
Αρχικά κάθε μια από τις 14 κορυφές του γραφήματος λαμβάνει ως ετικέτα ένα διαφορετικό τυχαίο αριθμό από το 1 εως το 14

Επιλέγουμε μια τυχαία σειρά στις κορυφές (διαλέγοντας μια τυχαία μετάθεση του [14]) πχ την μετάθεση 5 13 1 11 4 12 10 14 6 8 7 2 9 3.

Στην πρώτη επανάληψη, η κορυφή 5 λαμβάνει την ετικέτα 7, η κορυφή 13 λαμβάνει την ετικέτα 7, η κορυφή 1 λαμβάνει την ετικέτα 3, κορυφή 11 λαμβάνει την ετικέτα 9, η κορυφή 4 λαμβάνει την ετικέτα 7, η κορυφή 12 λαμβάνει την ετικέτα 9, η κορυφή 10 λαμβάνει την ετικέτα 9, η κορυφή 14 λαμβάνει την ετικέτα 3, η κορυφή 6 λαμβάνει την ετικέτα 8, η κορυφή 8 λαμβάνει την ετικέτα 8, η κορυφή 7 λαμβάνει την ετικέτα 7, η κορυφή 2 λαμβάνει την ετικέτα 3, κορυφή 9 λαμβάνει την ετικέτα 9, η κορυφή 3 λαμβάνει την ετικέτα 7.



Στην δεύτερη επανάληψη, οι κορυφές 1, 2, 14 λαμβάνουν την ετικέτα 7, οι κορυφές 6, 8 επίσης λαμβάνουν την ετικέτα 7, όλες οι υπόλοιπες κορυφές δεν αλλάζουν ετικέτα.



Στην τρίτη επανάληψη, δεν γίνεται καμία αλλαγή οπότε προκύπτει ότι στο γράφημα αυτό υπάρχουν 2 κοινότητες: Στην πρώτη ανήκουν οι κορυφές με ετικέτα 7 και στην δεύτερη ανήκουν οι κορυφές με ετικέτα 9.

Το πλεονέκτημα του αλγορίθμου διάδοσης ετικετών είναι ότι δεν απαιτεί εκ των προτέρων ούτε τον αριθμό των κοινοτήτων ούτε το μέγεθος κάθε μιας. Επίσης, δεν έχει κάποια παράμετρο ως είσοδο. Η υλοποίηση του είναι απλή και γρήγορη στην εκτέλεση ακόμα και σε γραφήματα με εκατομμύρια κορυφές. Τέλος, αν είναι γνωστό εκ των προτέρων ότι κάποιοι κόμβοι ανήκουν σε συγκεκριμένες κοινότητες, τότε οι ετικέτες τους μπορούν να χρησιμοποιηθούν για να κατασκευασθεί μια αρχική διαμέριση.

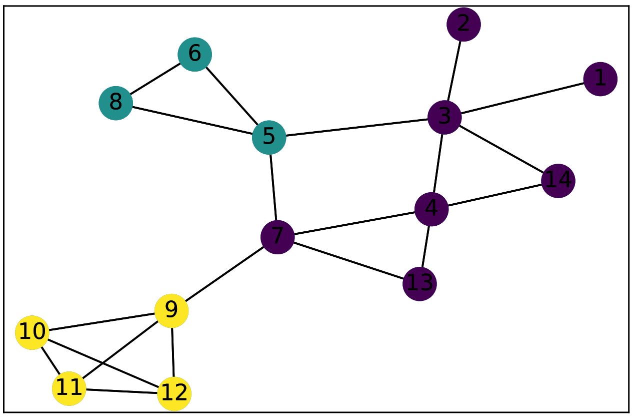
Η βιβλιοθήκη `networkx` διαθέτει μια υλοποίηση του αλγορίθμου διάδοσης ετικετών:

`partition = nx.community.asyn_lpa_communities(G)`

Ακολουθεί ένα παράδειγμα εκτέλεσης της μεθόδου, για το παραπάνω γράφημα, όπου ο αλγόριθμος ανιχνεύει 3 κοινότητες:

```
import networkx as nx
import matplotlib.pyplot as plt

G = nx.Graph()
n = 14 #A graph with 14 vertices
G.add_nodes_from(range(1,n+1))
E = [[1,3],[2,3],[3,4],[3,5],[3,14],[4,7],[4,13],[4,14],[5,7],[5,6],[5,8],
     [6,8],[7,9],[7,13],[9,10],[9,11],[9,12],[10,11],[10,12],[11,12]]
G.add_edges_from(E)

#get the communities using label propagation algorithm
partition = nx.community.asyn_lpa_communities(G)

#assign colors to nodes based on the communities
colors = [0 for i in range(1,n+2)]
color_id = 1
for community in partition:
    print("Comunity",color_id,"is:",community)
    for v in community: colors[v] = color_id
    color_id += 1 #assign different color to next community
colors.pop(0)

pos = nx.layout.kamada_kawai_layout(G)
nx.draw_networkx(G,pos,node_color=colors)
```

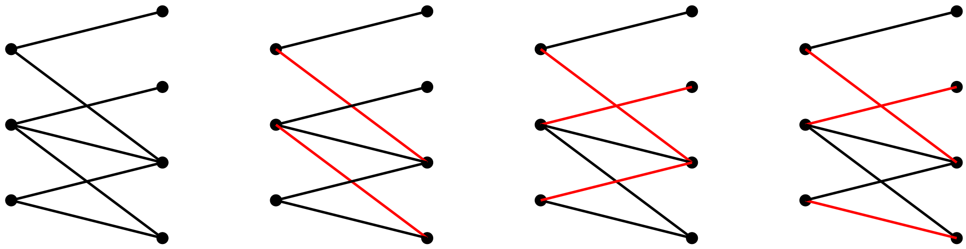
```
Comunity 1 is: {1, 2, 3, 4, 7, 13, 14}
Comunity 2 is: {8, 5, 6}
Comunity 3 is: {9, 10, 11, 12}
```



37. ΤΑΙΡΙΑΣΜΑΤΑ

Ένα σύνολο δεσμών $M \subseteq E$ ενός γραφήματος $G = (V, E)$ ονομάζεται **ταίριασμα** (matching) του G ανν δεν υπάρχουν δεσμοί στο M με κοινή κορυφή, ή ισοδύναμα κάθε κορυφή του G περιέχεται το πολύ σε ένα δεσμό του M.

Οι κόκκινοι δεσμοί των σχημάτων (β) και (δ) αποτελούν ταιριάσματα του διμερούς γραφήματος (α), ενώ οι κόκκινοι δεσμοί του σχήματος (γ) δεν είναι ταίριασμα διότι υπάρχουν δεσμοί που έχουν κοινό άκρο.



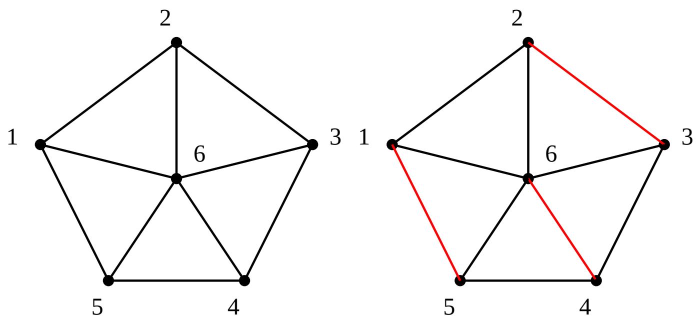
(α) (β) (γ) (δ)

Οι κορυφές του G που περιέχονται σε δεσμούς του ταιριάσματος M ονομάζονται **ταιριασμένες** (matched) κορυφές. Αν ο δεσμός $\{u, v\}$ ανήκει στο ταίριασμα M τότε λέμε ότι η κορυφή u **ταιριάζει** με την κορυφή v στο M. Μια κορυφή που δεν ανήκει σε δεσμό του M ονομάζεται **ακόρεστη** (unsaturated) ή **ελεύθερη** (free) (ως προς το M). Το πλήθος των δεσμών ενός ταιριάσματος M ονομάζεται **μέγεθος** του M και σημειώνεται με $|M|$

Ένα ταίριασμα M ονομάζεται **μέγιστο** (maximum) ανν έχει τον μεγαλύτερο δυνατό αριθμό δεσμών. Στο παράδειγμα, το ταίριασμα (δ) είναι μέγιστο, ενώ το ταίριασμα (β) δεν είναι μέγιστο.

Ένα ταίριασμα M ονομάζεται **τέλειο** (perfect) ανν κάθε κορυφή του γραφήματος G εμφανίζεται σε ακριβώς ένα δεσμό του M, ή ισοδύναμα ανν οι δεσμοί του M αποτελούν μια διαμέριση των κορυφών του G σε ζεύγη.[6]

Κανένα από τα ταιριάσματα (β), (δ) δεν είναι τέλειο. Στο σχήμα (στ) οι κόκκινοι δεσμοί αποτελούν ένα τέλειο ταίριασμα για το γράφημα (ε)

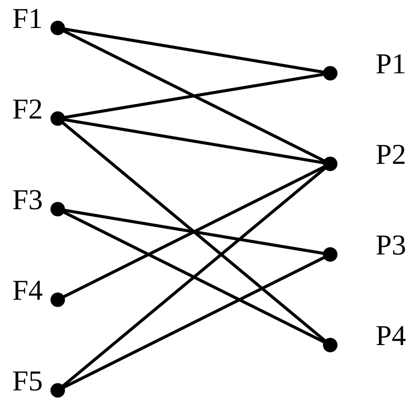


(ε) (στ)

Οι πιο συνηθισμένες εφαρμογές των ταιριασμάτων αναφέρονται σε διμερή γραφήματα. Ένα κλασικό παράδειγμα όπου μας ενδιαφέρει να βρούμε το μέγιστο ταίριασμα είναι το εξής: Έστω ένα διμερές γράφημα που αφορά το πρόγραμμα άσκησης ενός Τμήματος. Οι κορυφές του διμερούς γραφήματος αντιπροσωπεύουν

[6] Ένα τέλειο ταίριασμα ονομάζεται και 1-**παράγοντας** (1-factor) του G (οι δεσμοί του M αποτελούν ένα γεννητικό υπογράφημα του G του οποίου οι κορυφές έχουν βαθμό 1.)

αφενός τους φοιτητές και αφετέρου διαθέσιμες θέσεις πρακτικής άσκησης. Οι δεσμοί του ορίζονται με βάση τις προτιμήσεις των φοιτητών. Κάθε δεσμός συνδέει ένα φοιτητή με μια θέση που τον ενδιαφέρει. Ένας φοιτητής μπορεί να ενδιαφέρεται για πολλές θέσεις.

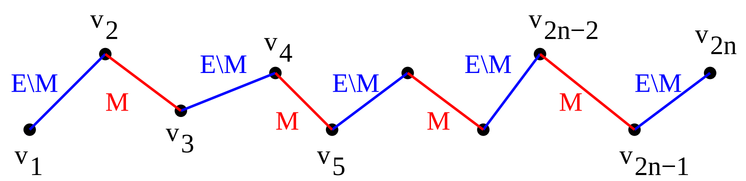


Ένα μέγιστο ταίριασμα στο γράφημα αυτό βρίσκει την πολυπληθέστερη ανάθεση, με τον μεγαλύτερο δυνατό αριθμό ενδιαφερόμενων να είναι ευχαριστημένοι.

Προφανώς, για κάθε ταίριασμα M ισχύει ότι $|M| \leq |E|$ και $|M| \leq |V|/2$. Επίσης, προφανώς, αν ένα γράφημα έχει περιττό αριθμό κορυφών, τότε δεν περιέχει τέλειο ταίριασμα.

Πως γνωρίζουμε αν ένα ταίριασμα M είναι μέγιστο; Ο Claude Berge έδωσε την απάντηση χρησιμοποιώντας την έννοια του M-αυξητικού μονοπατιού:

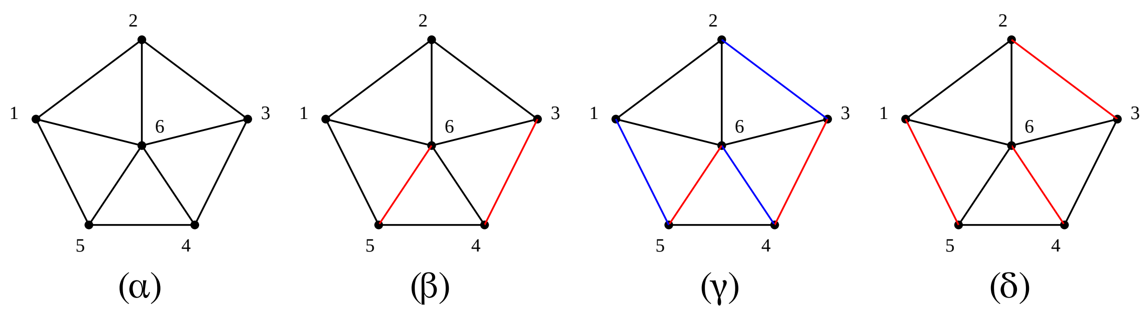
Ένα M-**αυξητικό** (M-augmenting) μονοπάτι στο G είναι ένα μονοπάτι P του οποίου οι δεσμοί ανήκουν εναλλάξ στα σύνολα $E \setminus M$ και M και τα άκρα του είναι ελεύθερες κορυφές.



Επομένως, το P θα έχει περιττό μήκος $2n-1$, $n \geq 1$ και άρα θα είναι της μορφής $P = (v_1, v_2, \ldots, v_{2n-1}, v_{2n})$ όπου οι n δεσμοί $\{v_1, v_2\}$, $\{v_3, v_4\}$, …, $\{v_{2i-1}, v_{2i}\}$, …, $\{v_{2n-1}, v_{2n}\}$ ανήκουν στο $E \setminus M$ και οι $n-1$ δεσμοί $\{v_2, v_3\}$, $\{v_4, v_5\}$, …, $\{v_{2i}, v_{2i+1}\}$, …, $\{v_{2n-2}, v_{2n-1}\}$ ανήκουν στο M.

Γενικότερα, ένα M-**εναλλασσόμενο** (M-alternating) μονοπάτι στο G είναι ένα μονοπάτι του οποίου οι δεσμοί ανήκουν εναλλάξ στα $E \setminus M$ και M.

Παράδειγμα : Το μονοπάτι $P = (2, 3, 4, 6, 5)$ του σχήματος (γ) είναι ένα M-αυξητικό μονοπάτι για το ταίριασμα $M = \{\{3, 4\}, \{5, 6\}\}$. του σχήματος (β).



Πρόταση 57 (Claude Berge, 1973). *Ένα ταίριασμα M του G είναι μέγιστο ανν το G δεν περιέχει M-αυξητικό μονοπάτι.*

Απόδειξη. Αν υπάρχει M-αυξητικό μονοπάτι για το M, τότε το μέγεθος του ταιριάσματος μπορεί να αυξηθεί, οπότε το ταίριασμα αυτό δεν είναι μέγιστο.

Αν δεν υπάρχει τέτοιο M-αυξητικό μονοπάτι τότε αυτό είναι μέγιστο. Πράγματι, έστω ότι αυτό δεν ισχύει για κάποιο συγκεκριμένο ταίριασμα M σε ένα γράφημα G. Έστω M^* ένα μέγιστο ταίριασμα στο G. Σύμφωνα με την υπόθεση ο αριθμός των δεσμών του M^* είναι μεγαλύτερος τουλάχιστον κατά 1 από τον αριθμό των δεσμών του M, δηλαδή $|M^*| > |M|$. Θεωρούμε τους δεσμούς στο ταίριασμα που προκύπτει από την συμμετρική διαφορά $M \triangle M^* = (M \setminus M^*) \cup (M^* \setminus M)$, δηλαδή στο σύνολο όλων των δεσμών που ανήκουν ή στο M, ή στο M^*, αλλά όχι και στα δύο. Παρατηρήστε ότι $|M^* \setminus M| > |M \setminus M^*|$, επειδή εξ υποθέσεως $|M^*| > |M|$. Έστω G' το υπογράφημα του G που αποτελείται από τους δεσμούς του $M \triangle M^*$. Από τον ορισμό του ταιριάσματος, μια οποιασδήποτε κορυφή που ανήκει στο $G' \subseteq G$ μπορεί να περιέχεται σε ένα το πολύ δεσμό του M και σε ένα το πολύ δεσμό του M^*. Επομένως, καθεμιά από τις κορυφές του G' έχει βαθμό 2 ή μικρότερο και επομένως κάθε συνεκτική συνιστώσα του G' είναι είτε ένα μονοπάτι, είτε ένας κύκλος αρτίου μήκους, αποτελούμενος από εναλλασσόμενους δεσμούς, από τα $M \setminus M'$ και $M^* \setminus M$. Εφόσον $|M^*| > |M|$ και ο αριθμός των δεσμών που προέρχονται από τα $M \setminus M^*$ και $M^* \setminus M$ είναι ο ίδιος για οποιονδήποτε κύκλο αρτίου μήκους με εναλλάξ ακμές στο G', θα πρέπει να υπάρχει τουλάχιστον ένα μονοπάτι αποτελούμενο από εναλλάξ ακμές, το οποίο να ξεκινά και να τερματίζεται ένα δεσμό από το $M^* \setminus M$. Επομένως, αυτό είναι ένα M-αυξητικό μονοπάτι για το ταίριασμα M, κάτι που έρχεται σε αντίθεση με την υπόθεση ότι δεν υπάρχει τέτοιο M-αυξητικό μονοπάτι. □

Η αλγοριθμική αξία της πρότασης του Berge είναι πολύ σημαντική. Αν ένα ταίριασμα δεν είναι μέγιστο τότε πάντα υπάρχει ένα M-αυξητικό μονοπάτι, και με βάση αυτό το μονοπάτι μπορούμε να κατασκευάσουμε ένα ταίριασμα M' με μεγαλύτερο μέγεθος. Συγκεκριμένα, αν M είναι ένα ταίριασμα και $P = (v_1, v_2, \ldots, v_{2n-1}, v_{2n})$ ένα M-αυξητικό μονοπάτι μήκους $2n-1$, τότε σβήνοντας τους $n-1$ δεσμούς $\{v_2, v_3\}$, $\{v_4, v_5\}$, …, $\{v_{2i}, v_{2i+1}\}$, …, $\{v_{2n-2}, v_{2n-1}\}$ ανήκουν στο M και προσθέτοντας τους υπόλοιπους n δεσμούς $\{v_1, v_2\}$, $\{v_3, v_4\}$, …, $\{v_{2i-1}, v_{2i}\}$, …, $\{v_{2n-1}, v_{2n}\}$ του $E \setminus M$ προκύπτει ένα ταίριασμα M' του G με ένα παραπάνω δεσμό από ότι το M και δύο επιπλέον ταιριασμένες κορυφές.

Παρατηρήστε ότι αν μια κορυφή v είναι ταιριασμένη στο M τότε με αυτήν την κατασκευή, η v παραμένει ταιριασμένη και στο M' (αλλά αν εμφανίζεται στο M-αυξητικό μονοπάτι αλλάζει η κορυφή με την οποία ταιριάζει).

Υπάρχουν διάφοροι αλγόριθμοι για την εύρεση του μέγιστου ταιριάσματος σε ένα γράφημα που χρησιμοποιούν M-αυξητικά μονοπάτια. Ειδικά για διμερή γραφήματα είναι πολύ γνωστός ο **αλγόριθμος των Hopcroft και Karp** ο οποίος έχει πολυπλοκότητα $O(\sqrt{|V|} \times |E|)$.

Στην συνέχεια δίδεται ο σκελετός ενός απλούστερου αλγόριθμου ο οποίος βρίσκει το μέγιστο ταίριασμα σε οποιοδήποτε γράφημα δεσμών και μπορεί να υλοποιηθεί με αναζήτηση σε βάθος ή αναζήτηση κατά πλάτος.

Αλγόριθμος εύρεσης των μέγιστων ταιριασμάτων.

Είσοδος: Ένα γράφημα δεσμών $G = (V, E)$

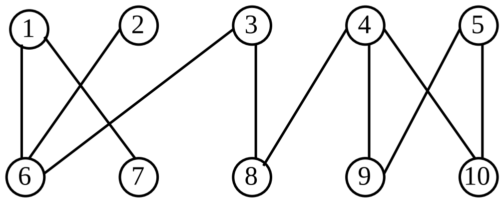
Έξοδος: Ένα μέγιστο ταίριασμα M του G

Η γενική περιγραφή του αλγορίθμου είναι η εξής:

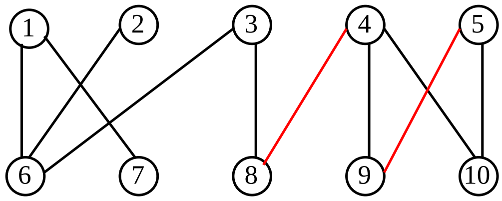
- Ξεκινάμε με ένα αρχικό ταίριασμα M (π.χ. κενό σύνολο δεσμών)
- (Επαναληπτικό βήμα) Βρίσκουμε ένα M-αυξητικό μονοπάτι και αυξάνουμε το τρέχον ταίριασμα M.
- Αν δεν μπορούμε να βρούμε τέτοιο μονοπάτι ο αλγόριθμος τερματίζεται και το ταίριασμα M που βρήκαμε είναι μέγιστο.

Σύμφωνα με την πρόταση του Berge αν το ταίριασμα δεν είναι μέγιστο υπάρχει πάντα ένα M-αυξητικό μονοπάτι. Πως βρίσκουμε ένα τέτοιο αυξητικό μονοπάτι P; Η ιδέα είναι απλή, ξεκινάμε από οποιαδήποτε ελεύθερη κορυφή v του G και χρησιμοποιώντας δεσμούς εναλλάξ μόνο από το $E \setminus M$ και το M προσπαθούμε να καταλήξουμε πάλι σε μια ελεύθερη κορυφή.

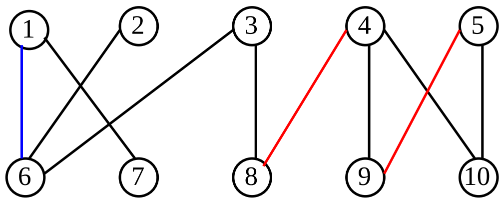
Παράδειγμα : Να βρεθεί ένα μέγιστο ταίριασμα για το διμερές γράφημα[7]



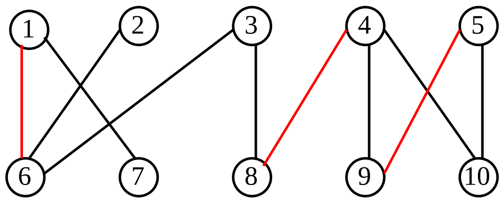
Δίδεται ως αρχικό ταίριασμα M το σύνολο των δεσμών $\{\{4, 8\}, \{5, 9\}\}$.



Όσο υπάρχουν ελεύθερες κορυφές, επιλέγουμε μια οποιαδήποτε ελεύθερη κορυφή, εδώ στο παράδειγμα την 1, και προσπαθούμε να κατασκευάσουμε ένα M-αυξητικό μονοπάτι P το οποίο καταλήγει σε ελεύθερη κορυφή. Οι γείτονες της 1 είναι οι κορυφές 6 και 7. Επιλέγουμε την κορυφή 6. Η κορυφή 6 είναι ελεύθερη, άρα βρήκαμε ένα M-αυξητικό μονοπάτι, το $P = (1, 6)$.

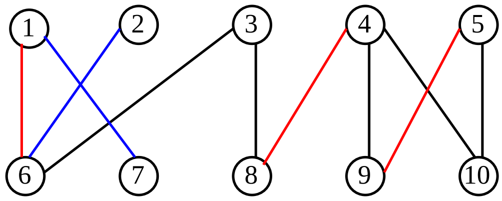


Οπότε προσθέτουμε τον δεσμό $\{1, 6\}$ στο M και προκύπτει το ταίριασμα $M = \{4, 8\}, \{5, 9\}, \{1, 6\}\}$.

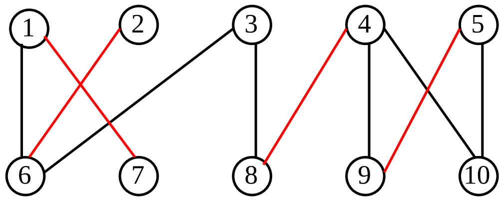


[7]Η μέθοδος λειτουργεί για οποιοδήποτε γράφημα

Επειδή υπάρχουν ελεύθερες κορυφές, επιλέγουμε μια από αυτές, εδώ στο παράδειγμα την 2, και προσπαθούμε πάλι να κατασκευάσουμε ενα M-αυξητικό μονοπάτι P (το οποίο καταλήγει σε ελεύθερη κορυφή). Ο μοναδικός γείτονας της 2 είναι η 6, η οποία είναι ταιριασμένη με την 1, άρα κατασκευάζουμε το M-εναλλασσόμενο μονοπάτι $P = (2, 6, 1)$. Συνεχίζουμε μέχρι να βρεθούμε σε αδιέξοδο ή να καταλήξουμε σε μια ελεύθερη κορυφή. Η κορυφή 1 έχει και άλλο γείτονα εκτός από την 6, την κορυφή 7. Η κορυφή 7 είναι ελεύθερη, άρα βρήκαμε ένα M-αυξητικό μονοπάτι, το $P = (2, 6, 1, 7)$.



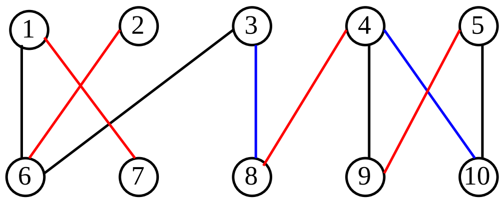
Οπότε αφαιρούμε τον δεσμό $\{16\}$ από το M και προσθέτουμε τους δεσμούς $\{2, 6\}$ και $\{1, 7\}$ και προκύπτει το ταίριασμα $M = \{\{4, 8\}, \{5, 9\}, \{2, 6\}, \{1, 7\}\}$.



Επειδή υπάρχουν ελεύθερες κορυφές, επιλέγουμε μια άλλη ελεύθερη κορυφή, εδώ στο παράδειγμα την 3, και προσπαθούμε να κατασκευάσουμε ένα M-αυξητικό μονοπάτι P. Η κορυφή 3 έχει δύο γείτονες, την κορυφή 6 και την κορυφή 8.

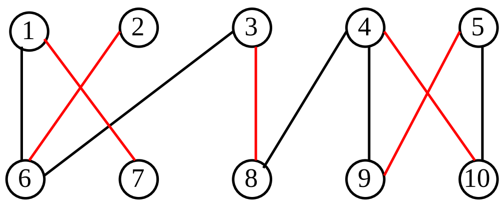
Επιλέγουμε την κορυφή 6 η οποία είναι ταιριασμένη με την κορυφή 2 οπότε το M-εναλλασσόμενο μονοπάτι P γίνεται $P = (3, 6, 2)$. Όμως η 2 δεν είναι δεν έχει άλλους γείτονες και δεν είναι ελεύθερη, οπότε αυτή η επιλογή οδηγεί σε αδιέξοδο.

Επιλέγουμε την κορυφή 8 η οποία είναι ταιριασμένη με την κορυφή 4 οπότε το M-εναλλασσόμενο μονοπάτι P γίνεται $P = (3, 8, 4)$. Η κορυφή 4 έχει δύο γείτονες, την κορυφή 9 και την 10. Επιλέγουμε την κορυφή 10 η οποία είναι ελεύθερη, άρα βρήκαμε ένα M-αυξητικό μονοπάτι, το $P = (3, 8, 4, 10)$.



Οπότε σβήνουμε τον δεσμό $\{8, 4\}$ από το M και προσθέτουμε τους δεσμούς $\{3, 8\}$ και $\{4, 10\}$ και προκύπτει το ταίριασμα

$$M = \{5, 9\}, \{2, 6\}, \{1, 7\}, \{3, 8\}, \{4, 10\}.$$



Επειδή δεν υπάρχουν ελεύθερες κορυφές (άρα δεν υπάρχουν ούτε M-αυξητικά μονοπάτια) ο αλγόριθμος ολοκληρώθηκε. Άρα το ταίριασμα

$$M = \{5, 9\}, \{2, 6\}, \{1, 7\}, \{3, 8\}, \{4, 10\}.$$

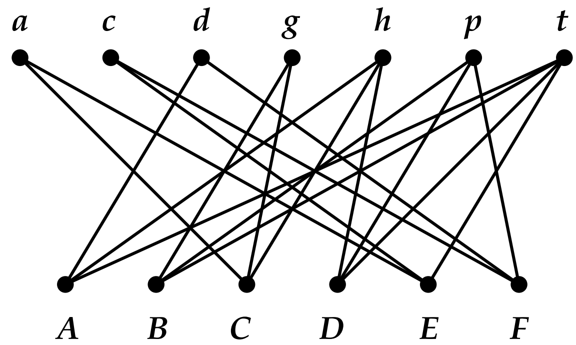
είναι ένα μέγιστο ταίριασμα του G, το οποίο εδώ στο παράδειγμα είναι και τέλειο.

Άσκηση 15. *Σε κάποιο project είναι προγραμματισμένες να γίνουν έξι εργασίες A, B, C, D, E, F. Για την εκτέλεση καθεμιάς από αυτές τις εργασίες είναι διαθέσιμοι οι παρακάτω συνεργάτες a, c, d, g, h, p, t. Κάθε εργασία μπορεί να διεκπεραιωθεί μόνο από ορισμένους συνεργάτες, σύμφωνα με τον παρακάτω πίνακα:*

A: d, h, t	*B: g, p, t*	*C: a, g, k*
D: h, p, t	*E: a, c, t*	*F: c, d, p*

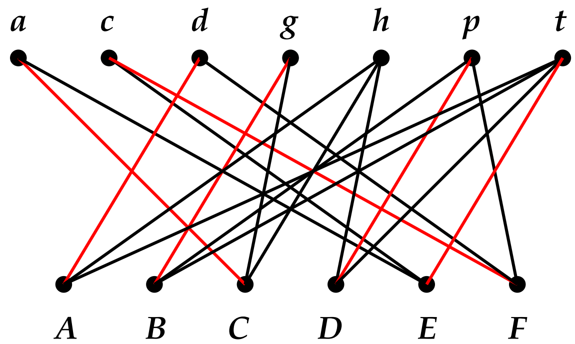
Πως πρέπει να γίνει η ανάθεση των εργασιών ώστε να καλυφθούν όλες οι εργασίες από ένα συνεργάτη η κάθε μία;

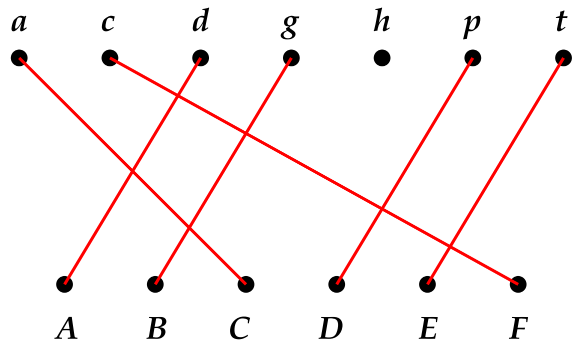
Λύση. Μπορούμε να μοντελοποιήσουμε το πρόβλημα χρησιμοποιώντας το επόμενο διμερές γράφημα:



Οι κορυφές του γραφήματος αποτελούνται αφενός από το σύνολο $U = \{A, B, C, D, E, F\}$ των εργασιών, και αφετέρου από το σύνολο $W = \{a, c, d, g, h, p, t\}$ των συνεργατών. Μια κορυφή $u \in U$ ενώνεται με μια κορυφή $w \in W$ αν ο u μπορεί να εκτελέση την εργασία w.

Μια λύση του προβλήματος δίδεται από το επόμενο μέγιστο ταίριασμα:





□

Τέλεια ταιριάσματα σε διμερή γραφήματα. Για κάθε υποσύνολο κορυφών $S \subseteq V$ του G, με $N(S)$ συμβολίζεται το σύνολο των κορυφών του G που είναι γειτονικές με τουλάχιστον μια κορυφή του S και ονομάζεται **γειτονιά** (neighbor) του S.

Πρόταση 58 (Philip Hall). *Έστω $G = (V, E)$ ένα διμερές γράφημα με διαμέριση κορυφών (U, W), και $|U| \leq |W|$. Τότε το G έχει ταίριασμα που περιέχει όλες τις κορυφές του U ανν*

$$|S| \leq |N(S)|$$

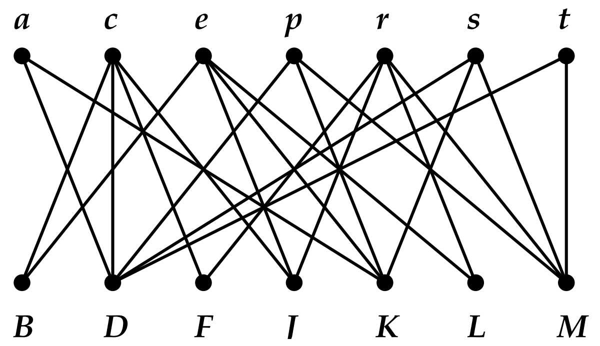
για κάθε υποσύνολο S του U.

Παράδειγμα : Επτά φοιτητές B, D, F, J, K, L, M ενδιαφέρονται να κάνουν πρακτική άσκηση σε επτά προσφερόμενες θέσεις a, c, e, p, r, s, t. Κάθε ένας από τους επτά έχει κάνει αίτηση σε κάποιες από αυτές:

B: c, e	D: a, c, p, s, t	F: c, r
J: c, e, r	K: a, e, p, s	L: e, r
M: p, r, s, t		

Είναι δυνατόν κάθε φοιτητής να κάνει πρακτική σε κάποια από τις θέσεις που τον ενδιαφέρουν;

Λύση. Το πρόβλημα αυτό μπορεί να μοντελοποιηθεί ως πρόβλημα εύρεσης τέλειου ταιριάσματος στο επόμενο διμερές γράφημα.



Οι κορυφές του γραφήματος αποτελούνται αφενός από το σύνολο $U = \{B, D, F, J, K, L, M\}$ των φοιτητών, και αφετέρου από το σύνολο $W = \{a, c, e, p, r, s, t\}$ των προσφερόμενων θέσεων. Μια κορυφή $u \in U$ ενώνεται με μια κορυφή $w \in W$ αν ο u έχει κάνει αίτηση για την θέση w.

Αν θεωρήσουμε το υποσύνολο $X = \{B, F, J, L\}$ έχουμε ότι $N(X) = \{c, e, r\}$. Επειδή $|N(X)| = 3 < 4 = |X|$ από το θεώρημα του Hall προκύπτει ότι δεν υπάρχει ταίριασμα για το διμερές γράφημα με μέγεθος 7. Επομένως, δεν είναι δυνατόν να γίνει αντιστοίχιση των επτά φοιτητών στις επτά θέσεις έτσι ώστε όλοι να είναι ευχαριστημένοι. □

Πόρισμα 59. *Ένα k-κανονικό διμερές γράφημα έχει τέλειο ταίριασμα.*

Απόδειξη. Έστω (U, W) μια διαμέριση των κορυφών του σύμφωνα με την ιδιότητα του διμερούς γραφήματος. Για κάθε υποσύνολο S του U υπάρχουν $k|S|$ δεσμοί τον οποίων ακριβώς το ένα άκρο ανήκει στο S. Το άλλο άκρο κάθε δεσμού ανήκει στην γειτονιά του S, η οποία είναι υποσύνολο του W, δηλαδή $N(S) \subseteq W$.

Επειδή από την γειτονιά $N(S)$ ξεκινούν τουλάχιστον $k|S|$ δεσμοί, έπεται ότι $k|N(S)| \geq k|S|$, δηλαδή $|N(S)| \geq |S|$. □

Τέλεια ταιριάσματα σε γραφήματα δεσμών. Μια συνεκτική συνιστώσα ενός γραφήματος ονομάζεται άρτια ή περιττή ανάλογα αν περιέχει άρτιο ή περιττό πλήθος κορυφών. Έστω $O(G)$ το πλήθος των περιττών συνεκτικών συνιστωσών του G.

Πρόταση 60 (William Tutte, 1947). *Ένα γράφημα $G = (V, E)$ έχει τέλειο ταίριασμα ανν*

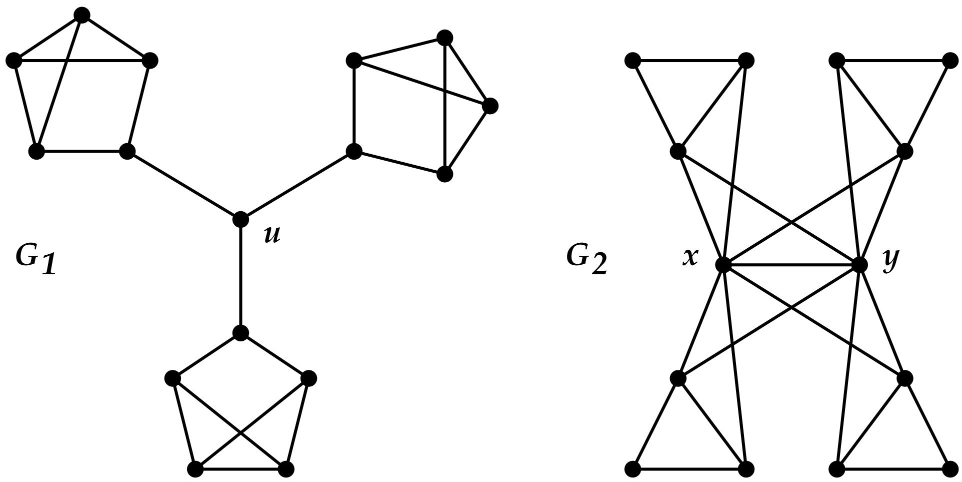
$$O(G - S) \leq |S|$$

για κάθε γνήσιο υποσύνολο S του V.

Με αντιθετοαναστροφή, το θεώρημα του Tutte μας λέει ότι αν υπάρχει κάποιο γνήσιο υποσύνολο S των κορυφών ενός γραφήματος G έτσι ώστε $O(G - S) > |S|$, τότε το G δεν έχει τέλειο ταίριασμα.

Έτσι, για παράδειγμα, αν το G έχει περιττό πλήθος κορυφών και S είναι το κενό σύνολο τότε $O(G - S) \geq 1 > 0 = |S|$, άρα είναι αδύνατο το G να έχει τέλειο ταίριασμα.

Πιο ενδιαφέροντα είναι τα επόμενα παραδείγματα: Τα γραφήματα G_1 και G_2 δεν έχουν τέλειο ταίριασμα.

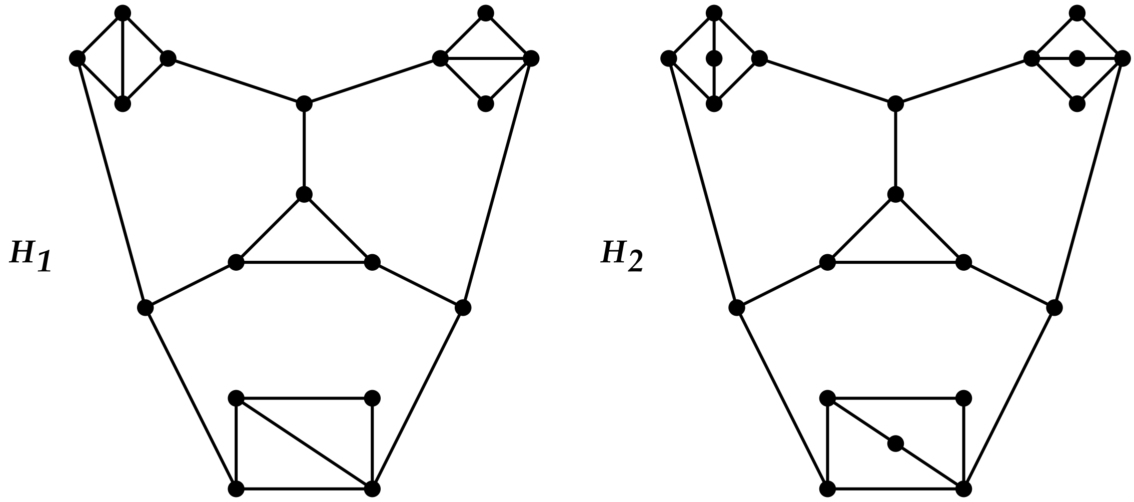


Πράγματι, αν $S_1 = \{u\}$, τότε $O(G_1 - S_1) = 3 > 1 = |S_1|$.

Επίσης, αν $S_2 = \{x, y\}$, τότε $O(G_2 - S_2) = 4 > 2 = |S_2|$.

Άρα, από το θεώρημα του Tutte τα G_1, G_2 δεν έχουν τέλεια ταιριάσματα.

Άσκηση 16. *Να εξετασθεί αν τα επόμενα γραφήματα έχουν τέλειο ταίριασμα ή όχι.*

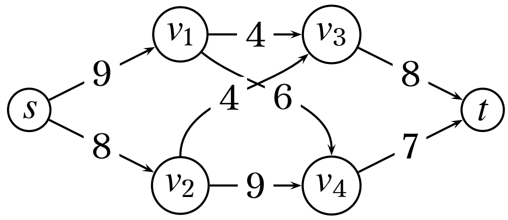


Πόρισμα 61 (Petersen, 1891). *Κάθε κυβικό γράφημα χωρίς γέφυρες έχει τέλειο ταίριασμα.*

38. ΔΙΚΤΥΑ ΡΟΗΣ

Δίκτυο ροής ή δίκτυο μεταφοράς ονομάζεται ένα κατευθυνόμενο γράφημα $G = (V, E)$ με βάρη στο οποίο υπάρχει μια κορυφή s η οποία ονομάζεται πηγή (source) και μια κορυφή t η οποία ονομάζεται καταβόθρα (sink). Το βάρος $c(e)$ ενός τόξου e χαρακτηρίζεται ως η χωρητικότητα (capacity) του τόξου e.

Παράδειγμα. Ένα δίκτυο ροής



Ροή f ενός δικτύου μεταφοράς ονομάζεται μια αντιστοίχιση μη αρνητικών τιμών στα τόξα του γραφήματος έτσι ώστε

- Η ροή $f(e)$ κάθε τόξου e δεν υπερβαίνει την χωρητικότητα του τόξου, δηλαδή

$$0 \leq f(e) \leq c(e), \text{ για κάθε } e \in E$$

- Για όλες τις κορυφές $v \in V$, (εκτός τις s και t), η ποσότητα ροής που εισέρχεται στην κορυφή v ισούται με την ποσότητα ροής που εξέρχεται από την v, δηλαδή

$$\sum_{(w,v)\in E} f(wv) = \sum_{(v,z)\in E} f(vz)$$

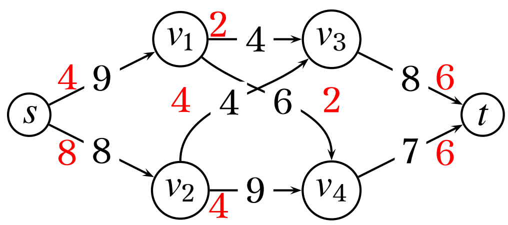
  με άλλα λόγια δεν υπάρχει απώλεια ροής.

Όταν ένα τόξο e διαθέτει ροή ίση με την χωρητικότητα του, τότε ονομάζεται κορεσμένο (saturated), ενώ σε αντίθετη περίπτωση ονομάζεται ακόρεστο.

Το μέγεθος ή ισχύς $|f|$ μιας ροής f ισούται με την συνολική ποσότητα που μεταφέρεται από την s στην t, το οποίο λόγω του ότι δεν υπάρχουν απώλειες ισούται με

$$|f| = \text{μέγεθος της } f = \sum_{(s,u)\in E} f(su)$$

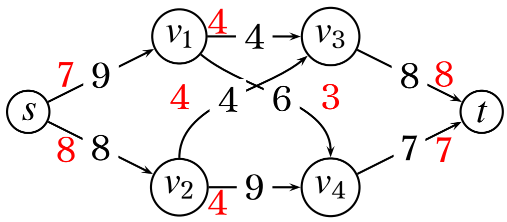
Παράδειγμα. Μια ροή με μέγεθος 12 για το παραπάνω δίκτυο μεταφοράς.



Τα δίκτυα ροής ή μεταφοράς χρησιμοποιούνται για την μοντελοποίηση δικτύων μεταφοράς αγαθών, όπως για παράδειγμα ένα δίκτυο ενέργειας (ΔΕΗ), ή ένα δίκτυο ύδρευσης, ή ένα τηλεπικοινωνιακό δίκτυο. Συνήθως σε αυτά τα δίκτυα υπάρχουν περιορισμοί στην ροή των αγαθών από σημείο σε σημείο. Οπότε κρίνεται αναγκαία η εύρεση της μέγιστης ροής που αντέχει το δίκτυο.

Μια ροή f ονομάζεται μέγιστη όταν έχει το μέγιστο δυνατό μέγεθος, δηλαδή δεν υπάρχει ροή f' με μεγαλύτερο μέγεθος.

Παράδειγμα. Μια μέγιστη ροή με μέγεθος 15 για το προηγούμενο δίκτυο μεταφοράς.



Η μέγιστη ροή σε ένα δίκτυο συνδέεται στενά με την έννοια της ελάχιστης τομής του δικτύου.

Τομή δικτύου (cut) ονομάζεται οποιαδήποτε διαμέριση του V σε δύο σύνολα V' και V'' με $s \in V'$ και $t \in V''$.

Παράδειγμα. Στο προηγούμενο δίκτυο είναι $V = \{s, v_1, v_2, v_3, v_4, t\}$. Μια τομή του G είναι $V = V' \cup V''$ όπου $V = \{s, v_1, v_2, v_4\}$ και $V = \{v_3, t\}$.

Η χωρητικότητα μιας τομής του $V = V' \cup V''$ ορίζεται ως το άθροισμα της χωρητικότητας των τόξων με αρχή στο σύνολο V' και πέρας στο σύνολο V''.

Παράδειγμα. Η χωρητικότητα της τομής του προηγούμενου δικτύου με $V = \{s, v_1, v_2, v_4\}$ και $V = \{v_3, t\}$ ισούται με

$$c((v_1, v_3)) + c((v_2, v_3)) + c((v_4, t)) = 4 + 4 + 7 = 15$$

Μια τομή ονομάζεται ελάχιστη όταν έχει την μικρότερη χωρητικότητα.

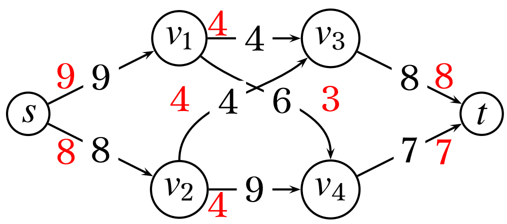
Πρόταση 62 (**Θεώρημα Ford - Fulkerson ή mincut - maxflow theorem**). Το μέγεθος της μέγιστης ροής σε κάθε δίκτυο μεταφοράς ισούται με την χωρητικότητα της ελάχιστης τομής του.

Υπάρχουν διάφοροι αλγόριθμοι για να υπολογίσουμε την μέγιστη ροή σε ένα δίκτυο μεταφοράς. Παρακάτω θα δοθεί ένας αλγόριθμος που βασίζεται στην έννοια της προροής.

Προροή f ενός δικτύου μεταφοράς ονομάζεται μια αντιστοίχιση τιμών στα τόξα του γραφήματος έτσι ώστε να μην παραβιάζονται οι περιορισμοί που θέτουν οι χωρητικότητες των τόξων και σε κάθε κορυφή η ροή εισόδου να είναι μεγαλύτερη ή ίση με την ροή εξόδου.

Όσες κορυφές έχουν περίσσευμα ροής, χαρακτηρίζονται ως ενεργές (active). Προφανώς, η πηγή και η καταβόθρα δεν θεωρούνται ενεργές.

Παράδειγμα. Μια προροή για το προηγούμενο δίκτυο μεταφοράς



Η κορυφή v_1 είναι ενεργή διότι έχει περίσσευμα ροής 2.

Προφανώς, με την εξάλειψη των ενεργών κορυφών, μια προροή μετατρέπεται σε "νόμιμη" ροή. Ο αλγόριθμος που θα παρουσιαστεί το πετυχαίνει με συστηματικές προωθήσεις της πλεονάζουσας ροής είτε προς την καταβόθρα είτε προς την πηγή.

Για την υλοποίηση του θα χρησιμοποιηθεί

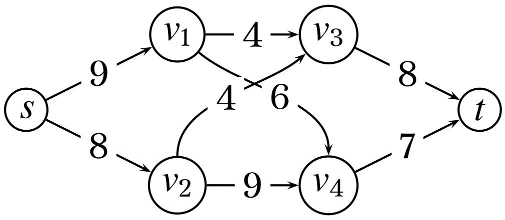
- ένας βοηθητικός πίνακας A με μέγεθος ίσο με το πλήθος των κορυφών του γραφήματος και με αρχικές τιμές την απόσταση κάθε κορυφής από την καταβόθρα, καθώς και
- μια ουρά προτεραιότητας[8] Q με κλειδιά τις τιμές του πίνακα A.

Αλγόριθμος προροής –προωθήσεως

(1) Αρχικά στις αντίστοιχες θέσεις του πίνακα A αποθηκεύεται η απόσταση κάθε κορυφής από την καταβόθρα t.
(2) Στη συνέχεια οδηγούνται σε κορεσμό όλα τα τόξα της πηγής s.
(3) Τοποθετούμε τις ενεργές κορυφές στην ουρά Q.
(4) Όσο η ουρά Q είναι μη κενή εκτελούνται εξής βήματα:
   - Αφαιρούμε από την ουρά Q την ενεργή κορυφή με το **μεγαλύτερο** κλειδί, έστω την v.
   - Βρίσκουμε τις γειτονικές κορυφές της v, δηλαδή τις κορυφές οι οποίες ανήκουν σε τόξο με αρχή ή πέρας την v.
   - Όσο η v παραμένει ενεργή εκτελούμε τα εξής βήματα:
     (Προωθούμε ή επιστρέφουμε περίσσευμα ροής σε κάποια από τις γειτονικές κορυφές της v με μικρότερο κλειδί μέχρις ότου η v πάψει να είναι ενεργή. Συγκεκριμένα:)
     – Αν όλες οι γειτονικές κορυφές της v έχουν μεγαλύτερο κλειδί, αυξάνουμε την αντίστοιχη τιμή στον πίνακα A μέχρι να γίνει ένα παραπάνω από κάποια γειτονική κορυφή.
     – Προωθούμε περίσσευμα ροής σε μια γειτονική κορυφή με μικρότερο κλειδί μέχρις ότου η v πάψει να είναι ενεργή ή μέχρις ότου το τόξο που τις συνδέει γίνει κορεσμένο.
     – Επιστρέφουμε περίσσευμα ροής σε μια γειτονική κορυφή με μικρότερο κλειδί μέχρις ότου η v πάψει να είναι ενεργή ή μέχρις ότου επιστρέψουμε όλη τη ροή.
   - Τοποθετούμε τις νέες ενεργές κορυφές στην ουρά Q.

[8]Οι ουρές προτεραιότητας είναι δομές δεδομένων που υποστηρίζουν τις παρακάτω 3 λειτουργίες: Έλεγχος αν η ουρά είναι κενή, Εισαγωγή νέου στοιχείου και Εξαγωγή στοιχείου. Σε κάθε στοιχείο που εισάγεται στην ουρά δίδεται και ένα κλειδί. Στις κλασικές ουρές το κλειδί είναι η σειρά με την οποία εισάγεται το στοιχείο. Η εξαγωγή στοιχείου γίνεται με βάση την τιμή του κλειδιού. Στις απλές ουρές εξάγεται το στοιχείο με το μικρότερο κλειδί (FiFo), στις στοίβες εξάγεται το στοιχείο με το μέγιστο κλειδί (LiFo).

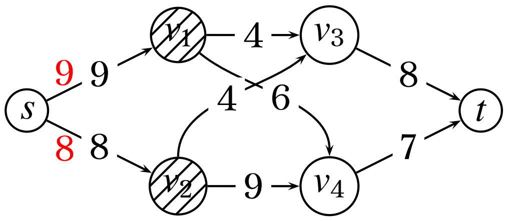
Παράδειγμα. Ο αλγόριθμος προροής–προωθήσεως στο προηγούμενο δίκτυο μεταφοράς εκτελείται ως εξής:



(1) Αρχικά θέτουμε σε κάθε κορυφή ως ύψος την απόσταση της από την καταβόθρα t

$$\begin{array}{rcl} & & \begin{array}{cccccc} s & 1 & 2 & 3 & 4 & t \end{array} \\ A &=& [\ \begin{array}{cccccc} 3, & 2, & 2, & 1, & 1, & 0, \end{array}\ ] \\ Q &=& [\ \qquad\qquad\qquad\qquad\qquad\ ] \end{array}$$

(2) Στη συνέχεια οδηγούμε σε κορεσμό όλα τα τόξα της πηγής.



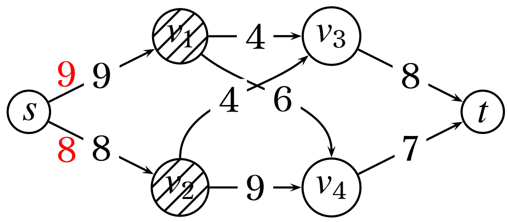
Οι κορυφές v_1 και v_2 είναι ενεργές, οπότε τοποθετούνται στην ουρά προτεραιότητας Q

$$\begin{array}{rcl} & & \begin{array}{cccccc} s & 1 & 2 & 3 & 4 & t \end{array} \\ A &=& [\ \begin{array}{cccccc} 3, & 2, & 2, & 1, & 1, & 0, \end{array}\ ] \\ Q &=& [\ \begin{array}{cccccc} & v_1 & v_2 & & & \end{array}\ ] \end{array}$$

(3) Επειδή η ουρά προτεραιότητας Q είναι μη κενή

$$\begin{array}{rcccccccl} & & s & 1 & 2 & 3 & 4 & t & \\ A = & [& 3, & 2, & 2, & 1, & 1, & 0, &] \\ Q = & [& & v_1 & v_2 & & & &] \end{array}$$

αφαιρούμε την κορυφή με το **μεγαλύτερο** κλειδί, υπάρχουν δύο επιλογές οι κορυφές v_1 και v_2, διαλέγουμε τυχαία την v_1.



Αφαιρούμε την v_1 από την ουρά προτεραιότητας Q. Η v_1 έχει περίσσευμα ροής 9.

Ισχύει ότι $A[1] = 2$

Οι γειτονικές κορυφές της v_1 είναι οι s, v_3 και v_4.

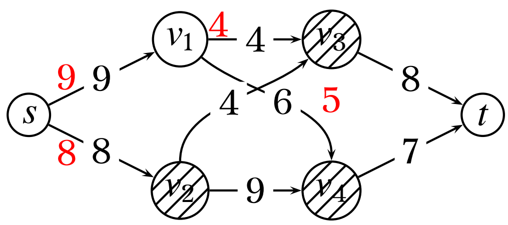
Ισχύει ότι $A[s] = 3$, $A[3] = 1$ και $A[4] = 1$.

Οι v_3 και v_4 έχουν μικρότερο κλειδί, επομένως θα προωθήσουμε το περίσσευμα ροής σ' αυτές.

Στην v_3 προωθούμε ροή ίση με την μέγιστη χωρητικότητα του τόξου που τις συνδέει, δηλαδή ίση με 4. Απομένει περισσευμα ροής $9 - 4 = 5$.

Στην v_4 προωθούμε ροή ίση με 5, (που είναι μικρότερη από την χωρητικότητα 6 του τόξου που τις συνδέει).

Επομένως, η v_1 παύει να είναι ενεργή.



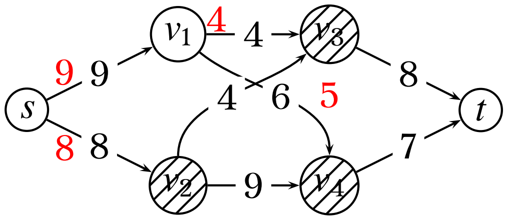
Οι κορυφές v_3 και v_4 γίνονται ενεργές, επομένως τις τοποθετούμε στην ουρά προτεραιότητας Q.

$$\begin{array}{rcccccccl} & & s & 1 & 2 & 3 & 4 & t & \\ A = & [& 3, & 2, & 2, & 1, & 1, & 0, &] \\ Q = & [& & & v_2 & v_3 & v_4 & &] \end{array}$$

(4) Επειδή η ουρά προτεραιότητας είναι μη κενή

$$\begin{array}{rccccccl} & s & 1 & 2 & 3 & 4 & t & \\ A = [& 3, & 2, & 2, & 1, & 1, & 0, &] \\ Q = [& & & v_2 & v_3 & v_4 & &] \end{array}$$

αφαιρούμε την κορυφή με το **μεγαλύτερο** κλειδί, δηλαδή την κορυφή v_2.



Αφαιρούμε την v_2 από την ουρά προτεραιότητας Q. Η v_2 έχει περίσσευμα ροής 8.

Ισχύει ότι $A[2] = 2$

Οι γειτονικές κορυφές της v_2 είναι οι s, v_3 και v_4.

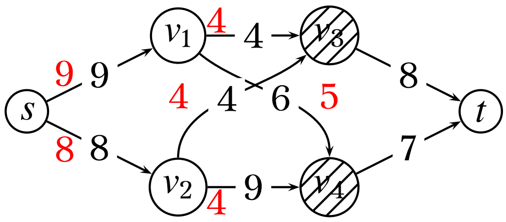
Ισχύει ότι $A[s] = 3$, $A[3] = 1$ και $A[4] = 1$.

Οι v_3 και v_4 έχουν μικρότερο κλειδί, επομένως θα προωθήσουμε το περίσσευμα ροής σ' αυτές.

Στην v_3 προωθούμε ροή ίση με την μέγιστη χωρητικότητα του τόξου που τις συνδέει, δηλαδή ίση με 4. Απομένει περισσευμα ροής $8 - 4 = 4$.

Στην v_4 προωθούμε ροή ίση με 4, (που είναι μικρότερη από την χωρητικότητα 9 του τόξου που τις συνδέει).

Επομένως, η v_2 παύει να είναι ενεργή.



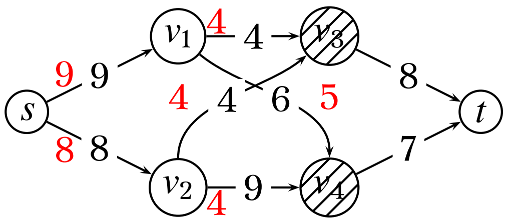
Οι ενεργές κορυφές v_3 και v_4 είναι ήδη μέσα στην ουρά Q.

$$\begin{array}{rccccccl} & s & 1 & 2 & 3 & 4 & t & \\ A = [& 3, & 2, & 2, & 1, & 1, & 0, &] \\ Q = [& & & & v_3 & v_4 & &] \end{array}$$

(5) Επειδή η ουρά προτεραιότητας Q είναι μη κενή

$$\begin{array}{rcccccccl} & & s & 1 & 2 & 3 & 4 & t & \\ A & = [& 3, & 2, & 2, & 1, & 1, & 0, &] \\ Q & = [& & & & v_3 & v_4 & &] \end{array}$$

αφαιρούμε την κορυφή με το **μεγαλύτερο** κλειδί, υπάρχουν δύο επιλογές οι κορυφές v_3 και v_4, διαλέγουμε τυχαία την v_3.



Αφαιρούμε την v_3 από την ουρά προτεραιότητας Q. Η v_3 έχει περίσσευμα ροής 8.

Ισχύει ότι $A[3] = 1$

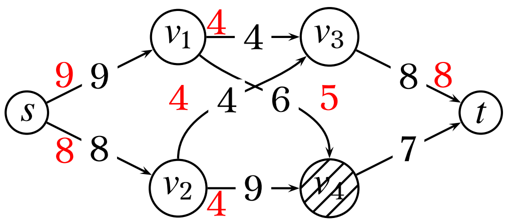
Οι γειτονικές κορυφές της v_3 είναι οι v_1, v_2 και t.

Ισχύει ότι $A[1] = 2$, $A[2] = 2$ και $A[t] = 0$.

Η t έχει μικρότερο κλειδί, επομένως θα προωθήσουμε το περίσσευμα ροής σ' αυτή.

Στην t προωθούμε ροή ίση με την μέγιστη χωρητικότητα του τόξου που τις συνδέει, δηλαδή ίση με 8. Απομένει περισσευμα ροής $8 - 8 = 0$.

Επομένως, η v_3 παύει να είναι ενεργή.



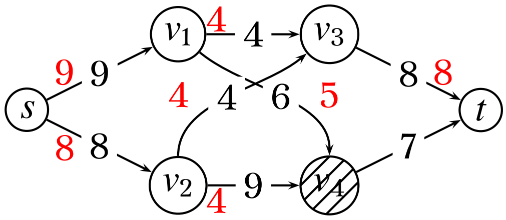
Η t ως καταβόθρα ποτέ δεν γίνεται ενεργή.

$$\begin{array}{rcccccccl} & & s & 1 & 2 & 3 & 4 & t & \\ A & = [& 3, & 2, & 2, & 1, & 1, & 0, &] \\ Q & = [& & & & & v_4 & &] \end{array}$$

(6) Επειδή η ουρά προτεραιότητας Q είναι μη κενή

$$\begin{array}{rcccccccl} & & s & 1 & 2 & 3 & 4 & t & \\ A & = [& 3, & 2, & 2, & 1, & 1, & 0, &] \\ Q & = [& & & & & v_4 & &] \end{array}$$

αφαιρούμε την κορυφή με το **μεγαλύτερο** κλειδί, δηλαδή την κορυφή v_4.



Αφαιρούμε την v_4 από την ουρά προτεραιότητας Q. Η v_4 έχει περίσσευμα ροής 9.

Ισχύει ότι $A[4] = 1$

Οι γειτονικές κορυφές της v_4 είναι οι v_1, v_2 και t.

Ισχύει ότι $A[1] = 2$, $A[2] = 2$ και $A[t] = 0$.

Η t έχει μικρότερο κλειδί, επομένως θα προωθήσουμε το περίσσευμα ροής σ' αυτή.

Στην t προωθούμε ροή ίση με την μέγιστη χωρητικότητα του τόξου που τις συνδέει, δηλαδή ίση με 7. Απομένει περισσευμα ροής $9 - 7 = 2$.

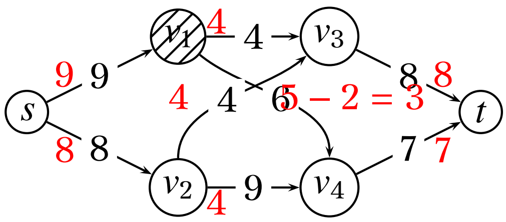
Επομένως, η v_4 παραμένει ενεργή, χωρίς να μπορούμε να προωθήσουμε στην t άλλη ροή.

Αυξάνουμε την τιμή $A[4]$ μέχρι ότου γίνει μεγαλύτερη από κάποια από τις γειτονικές της στην οποία μπορούμε να προωθήσουμε ροή, επειδή οι γειτονικές της έχουν τιμή 2 θέτουμε $A[4] = 3$.

Τώρα, οι v_1 και v_2 έχουνε μικρότερο κλειδί, επομένως θα προωθήσουμε το περίσσευμα ροής σ' αυτές.

Στην v_1 προωθούμε ροή ίση με το περίσσευμα της v_4 δηλαδή ίση με 2 (που είναι μικρότερη από την ροή 5 του τόξου που τις συνδέει).

Η v_4 παύει να είναι ενεργή.



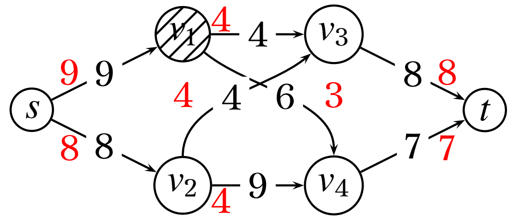
Η v_1 έχει τώρα περίσσευμα $9-(4+3) = 2$, οπότε είναι ενεργή και εισάγεται στην ουρά Q.

$$\begin{array}{rcccccccl} & & s & 1 & 2 & 3 & 4 & t & \\ A & = [& 3, & 2, & 2, & 1, & 3, & 0, &] \\ Q & = [& & v_1 & & & & &] \end{array}$$

(7) Επειδή η ουρά προτεραιότητας Q είναι μη κενή

$$\begin{array}{cc} & \begin{array}{cccccc} s & 1 & 2 & 3 & 4 & t \end{array} \\ A = & [\begin{array}{cccccc} 3, & 2, & 2, & 1, & 3, & 0, \end{array}] \\ Q = & [\begin{array}{cccccc} & v_1 & & & & \end{array}] \end{array}$$

αφαιρούμε την κορυφή με το **μεγαλύτερο** κλειδί, δηλαδή την κορυφή v_1.



Αφαιρούμε την v_1 από την ουρά προτεραιότητας Q. Η v_1 έχει περίσσευμα ροής 2.

Ισχύει ότι $A[1] = 2$

Οι γειτονικές κορυφές της v_4 είναι οι s, v_3 και v_4.

Ισχύει ότι $A[s] = 3$, $A[3] = 1$ και $A[4] = 3$.

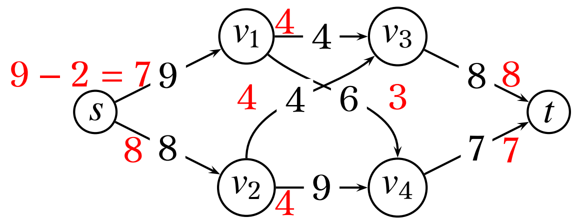
Η v_3 έχει μικρότερο κλειδί, αλλά δεν μπορούμε να προωθήσουμε άλλη ροή σ' αυτή.

Αυξάνουμε την τιμή $A[1]$ μέχρι ότου γίνει μεγαλύτερη από κάποια από τις γειτονικές της στην οποία μπορούμε να προωθήσουμε ροή, επειδή οι γειτονικές της έχουν τιμή 3 θέτουμε $A[1] = 4$.

Τώρα, οι s και v_4 έχουνε μικρότερο κλειδί, επομένως θα προωθήσουμε το περίσσευμα ροής σ' αυτές.

Στην s προωθούμε ροή ίση με το περίσσευμα της v_1 δηλαδή ίση με 2 (που είναι μικρότερη από την ροή 9 του τόξου που τις συνδέει).

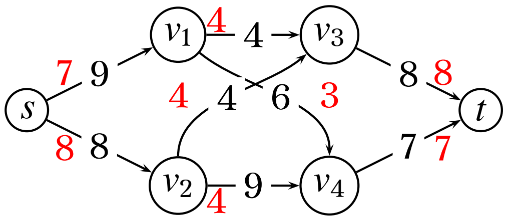
Η v_1 παύει να είναι ενεργή.



Η v ως καταβόθρα ποτέ δεν γίνεται ενεργή.

$$\begin{array}{cc} & \begin{array}{cccccc} s & 1 & 2 & 3 & 4 & t \end{array} \\ A = & [\begin{array}{cccccc} 3, & 2, & 2, & 1, & 3, & 0, \end{array}] \\ Q = & [\begin{array}{cccccc} & & & & & \end{array}] \end{array}$$

(8) Επειδή η ουρά προτεραιότητας Q είναι κενή, δεν υπάρχουν ενεργές κορυφές, επομένως βρέθηκε η μέγιστη ροή με μέγεθος ίσο με $7 + 8 = 15$.



Παρατηρήσεις

(1) (**Πολυπλοκότητα**) Αποδεικνύεται ότι ο αλγόριθμος προροής - προωθήσεως τερματίζει και ότι στην χειρότερη περίπτωση χρειάζεται χρόνο $O(|V|^2|E|)$. Ο τερματισμός οφείλεται στην ύπαρξη άνω φράγματος για τις τιμές του πίνακα A (και όχι στον προσανατολισμό των τόξων).

(2) (**Ιδιότητα ακεραιότητας**) Αποδεικνύεται ότι αν όλες οι χωρητικότητες είναι ακέραιες, τότε και η μέγιστη ροή θα έχει ακέραια τιμή (η εκτέλεση του αλγορίθμου προροής - προωθήσεως θα έχει ως αποτέλεσμα όλες οι ενδιάμεσες ροές πάνω στα τόξα να είναι επίσης ακέραιες).

(3) (**Μηχανικό ανάλογο**) Υπάρχει ένα μηχανικό ανάλογο για τον αλγόριθμο προροής – προωθήσεως:

Κάθε κορυφή v_i θεωρείται ότι είναι σε ύψος $A[i]$ από το έδαφος, ενώ τα τόξα είναι εύκαμπτοι πλαστικοί σωλήνες.

Αντικειμενικός σκοπός είναι να διοχευτεί νερό από την πηγή προς την καταβόθρα. Η διοχεύτεση γίνεται με τη βοήθεια της βαρύτητας, από την υψηλότερη προς την χαμηλότερη κορυφή.

Αρχικά, η πηγή υψώνεται και το νερό κυλάει προς τους γειτονές της.

Στη γενική περίπτωση θα μετακινηθεί προς την καταβόθρα. Μερικές φορές, όμως, παγιδεύεται τοπικά σε μια κορυφή, η οποία δεν διαθέτει χαμηλότερους γείτονες. Οπότε, αναγκαστικά, την υψώνουμε και η ροή προς την καταβόθρα αποκαθίσταται.

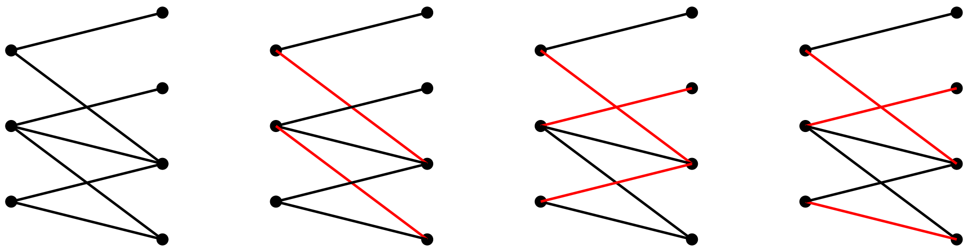
Τελικά, φτάνουμε σε σημείο που δεν μπορεί να διοχετευτεί επιπλέον ροή προς την καταβόθρα, οπότε καθώς οι κορυφές συνεχίζουν να ανυψώνονται, η πλεονάζουσα ροή αρχίζει να επιστρέφει πίσω στην πηγή.

Όταν τερματίζει η διαδικασία στο σύστημα έχει επέλθει ισορροπία.

Η σημαντικότητα των δικτύων ροής οφείλεται όχι μόνο στην προφανή χρησιμότητά τους αλλά και στο γεγονός ότι μοντελοποιούν (και επιλύουν) πολλά άλλα αλγοριθμικά προβλήματα τα οποία σε πρώτη ματιά δεν φαίνεται ότι σχετίζονται με δίκτυα ροής.

Παράδειγμα 1. Εύρεση μέγιστου ταιριάσματος σε διμερές γράφημα.
Ταίριασμα (matching) ενός διμερούς γράφηματος $G = (V_1 \cup V_2, E)$ ονομάζεται ένα υποσύνολο των δεσμών του G οι οποίοι ανά δύο δεν έχουν κοινά άκρα.

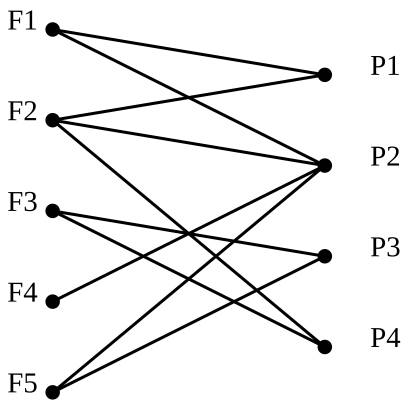
Οι κόκκινοι δεσμοί των σχημάτων (β) και (δ) αποτελούν ταιριάσματα του διμερούς γραφήματος (α), ενώ οι κόκκινοι δεσμοί του σχήματος (γ) δεν είναι ταίριασμα διότι υπάρχουν δεσμοί που έχουν κοινό άκρο.



(α) (β) (γ) (δ)

Ένα ταίριασμα ονομάζεται **μέγιστο** ανν περιέχει τον μέγιστο αριθμό δεσμών. Στο παράδειγμα, το ταίριασμα (δ) είναι μέγιστο, ενώ το ταίριασμα (β) δεν είναι μέγιστο.

Ένα κλασικό παράδειγμα όπου μας ενδιαφέρει να βρούμε το μέγιστο ταίριασμα είναι το εξής: Έστω ένα διμερές γράφημα που αφορά το πρόγραμμα άσκησης ενός Τμήματος. Οι κορυφές του διμερούς γραφήματος αντιπροσωπεύουν αφενός τους φοιτητές και αφετέρου διαθέσιμες θέσεις πρακτικής άσκησης. Οι δεσμοί του ορίζονται με βάση τις προτιμήσεις των φοιτητών. Κάθε δεσμός συνδέει ένα φοιτητή με μια θέση που τον ενδιαφέρει. Ένας φοιτητής μπορεί να ενδιαφέρεται για πολλές θέσεις.

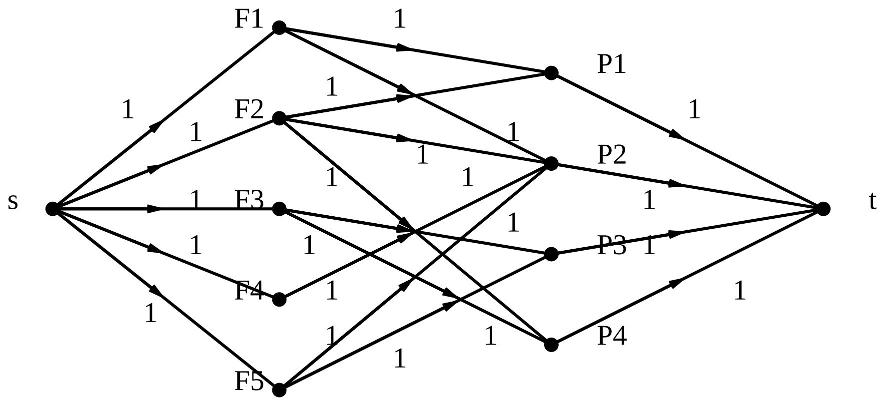


Ένα μέγιστο ταίριασμα στο γράφημα αυτό βρίσκει την πολυπληθέστερη ανάθεση, με όλους τους ενδιαφερόμενους ευχαριστημένους.

Το πρόβλημα του μέγιστου ταιριάσματος ενός διμερούς γραφήματος $G = (V_1 \cup V_2, E)$ μπορεί να αναχθεί σε πρόβλημα μέγιστης ροής ως εξής:

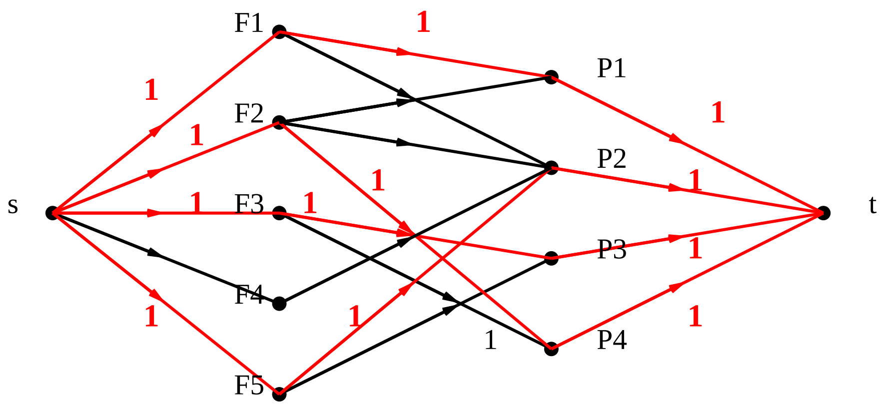
(1) Προσθέτουμε δύο νέες κορυφές: Την πηγή s και την καταβόθρα t.
(2) Για κάθε κορυφή v του V_1 προσθέτουμε το τόξο (s, v).
(3) Για κάθε κορυφή v' του V_2 προσθέτουμε το τόξο (v, t).
(4) Προσανατολίζουμε όλους τους δεσμούς του G από το V_1 προς το V_2.
(5) Σε κάθε τόξο e αναθέτουμε χωρητικότητα ίση με την μονάδα.

Στο παράδειγμα της πρακτικής άσκησης το δίκτυο ροής που δημιουργείται είναι το εξής:



Όταν βρούμε μια ροή στο νέο γράφημα μπορούμε να βρούμε και ένα ταίριασμα στο G αρκεί να κρατήσουμε όλους τους "κεντρικούς" δεσμούς που έχουν ροή. Πράγματι, επειδή κάθε κορυφή του V_1 διαθέτει μόνο μια εισερχόμενη ακμή χωρητικότητας μιας μονάδας θα συνδέεται μόνο με μια κορυφή του V_2 και αντιστρόφως. Άρα, το πρόβλημα του μέγιστου ταιριάσματος ανάγεται στο πρόβλημα της μέγιστης ροής.

Στο παράδειγμα, της πρακτικής άσκησης, η μέγιστη ροή έχει μέγεθος 4 όπως φαίνεται στο επόμενο σχήμα



και αντιστοιχεί στο μέγιστο ταίριασμα: (F_1, P_1), (F_2, P_4), (F_3, P_3) και (F_5, P_2).

Παρατήρηση. Με αυτή την μοντελοποίηση αποδεικνύεται ότι το κόστος εύρεσης του μέγιστου ταιριάσματος σε ένα διμερές γραφημα $G = (V_1 \cup V_2, E)$ είναι $O((|V_1| + |V_2|)|E|)$.

Παράδειγμα 2. **Εύρεση μέγιστου αριθμού ξένων (ως προς τις ακμές) μονοπατιών ανάμεσα σε δύο κορυφές**.

Έστω G ένα γράφημα δεσμών και u, v κορυφές του που ανήκουν στην ίδια συνεκτική συνιστώσα. Η συνεκτικότητα ακμών των u, v ορίζεται ως ο ελάχιστος αριθμός δεσμών που πρέπει να αφαιρέσουμε από το γράφημα G ώστε να τα u, v να μην συνδέονται με κάποιο μονοπάτι. Εύκολα μπορούμε να δούμε ότι ο αριθμός αυτός ισούται με τον μέγιστο αριθμό μονοπατιών που συνδέουν τους u, v και δεν έχουν κοινές ακμές, τα οποία ονομάζονται **ξένα (ως προς τις ακμές) μονοπάτια**. Αν υπάρχουν k τέτοια ξένα μονοπάτια τότε πρέπει και αρκεί να αφαιρέσουμε τουλάχιστον ένα δεσμό από κάθε μονοπάτι.

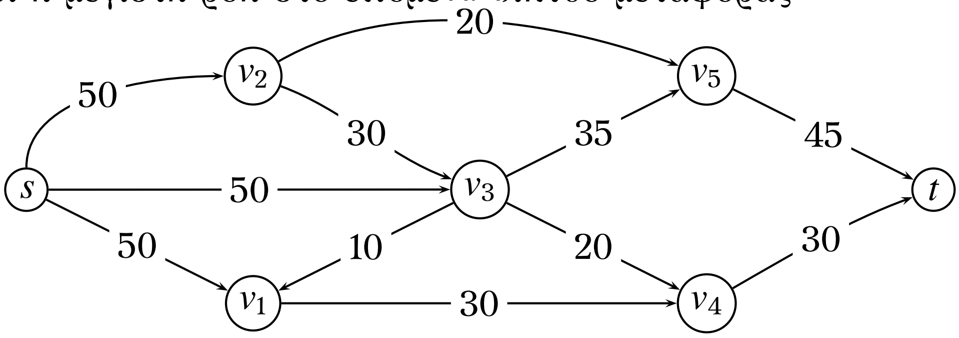
Το πρόβλημα της εύρεσης του μέγιστου αριθμού ξένων (ως προς τις ακμές) μονοπατιών ανάμεσα στις κορυφές u και v ενός γραφήματος δεσμών G μπορεί να αναχθεί σε πρόβλημα μέγιστης ροής ως εξής:

(1) Ορίζουμε ως πηγή s την κορυφή u και ως καταβόθρα καταβόθρα t την κορυφή v.
(2) Σε κάθε βήμα του αλγορίθμου ο προσανατολισμός κάθε τόξου καθορίζεται με βάση την υπάρχουσα ροή πάνω σ' αυτό. Αν δεν υπάρχει ροή τότε προσανολίζεται από την ενεργή κορυφή προς τον γείτονά της.
(3) Σε κάθε τόξο e αναθέτουμε χωρητικότητα ίση με την μονάδα.

Επειδή όλες οι χωρητικότητες είναι ίσες με 1 σε κάθε ενδιάμεσο τόξο πάνω στο οποίο υπάρχει ροή θα αντιστοιχεί ένα ακριβώς ένα τόξο που αρχίζει από το τέλος του και έχει ροή 1. Το μέγεθος της μέγιστης ροής θα αντιστοιχεί στο πλήθος όλων αυτών των ξένων μονοπατιών.

Ασκήσεις προς επίλυση

(1) Να βρεθεί η μέγιστη ροή στο επόμενο δίκτυο μεταφοράς



(2) Έστω ότι διευθύνετε ένα γραφείο συνοικεσίων. Περιγράψτε έναν αλγόριθμο, ο οποίος θα βρίσκει το μέγιστο αριθμό ιδανικών ζευγαριών μεταξύ ενός συνόλου p ανδρών και ενός συνόλου q γυναικών.

Βιβλιογραφία

[1] Λ. Κυρούσης, Χ, Μπούρας, Π. Σπυράκης, Γ. Σταματίου, Εισαγωγή στους Γράφους, Εκδόσεις CTI Press, Αθήνα, 1999.

[2] Π. Μποζάνης, Αλγόριθμοι, Εκδόσεις Τζιόλα, 2005.

[3] Σ. Νικολόπουλος, Αλγοριθμική Θεωρία Γραφημάτων, Κάλλιπος, 2015.

[4] Α. Παναγιωτόπουλος, Εισαγωγή στα Γραφήματα, Εκδόσεις Α. Σταμούλης, Πειραιάς, 1989.

[5] Θ. Παπαθεοδώρου, Αλγόριθμοι, Εκδόσεις Πανεπιστημίου Πατρών, Πάτρα, 1999.

[6] Α. Παπαϊωάννου, Θεωρία Γραφημάτων, Αθήνα, 2002.

[7] R. Balakrishnan and K. Ranganathan, A Textbook of Graph Theory, Springer, 2012.

[8] C. Berge, Théorie des Graphes et ses Applications, Dunod, Paris, 1958.

[9] A. Benjamin, G. Chartrand and P. Zhang, The Fascinating World of Graph Theory, Princeton University Press, 2015.

[10] B. Bollobás, Modern Graph Theory, Springer-Verlag, New York, 1998.

[11] M. Bona, A Walk Through Combinatorics: An introduction to Enumeration and Graph Theory, World Scientific, 2017.

[12] J. A. Bondy and U. S. R. Murty, Graph Theory, Springer, 2008.

[13] N. Dershowitz and S. Zaks, Enumerations of ordered trees, *Discrete Mathematics*, **31** (1980), 9–28.

[14] E. Deutsch, A bijection on ordered trees and its consequences, *Journal of Combinatorial Theory, Series A*, **90** (2000), 210–215.

[15] E. Deutsch, Ordered trees with prescibed root degrees, node degrees, and branch lenghts, *Discrete Mathematics*, **282** (2004), 89–94.

[16] R. Diestel, Graph Theory, Springer-Verlag, New York, 2017.

[17] J.-C. Fournier, Graph Theory and Applications, Wiley, 2006.

[18] F. Harary, Graph Theory, Addison-Wesley Publishing Co, Reading, Massachusetss, 1972.

[19] J. Harris, J. Hirst, M. Mossinghoff, Combinatorics and Graph Theory, Springer-Verlag, New York, 2000.

[20] W. D. Joyner and C. Grant Melles, Adventures in Graph Theory, Birkhauser, 2017.

[21] D. Knuth, The Art of Computer Programming, Vol. 1, *Fundamental Algorithms*, Addison-Wesley, second edition, 1973.

[22] D. Marcus, Graph Theory: A Problem Oriented Approach, Mathematical Association of America, 2008.

[23] Md. Saidur Rahman, Basic Graph Theory, Springer, 2017.

[24] S. Saha Ray, Graph Theory with Algorithms and its Applications, Springer, 2013.
[25] K. Thulasiraman, Handbook of Graph Theory, Combinatorial Optimization, and Algorithms, CRC Press, 2016.