Οι διαφάνειες βασίζονται σε αυτές του ακόλουθου μαθήματος:

Introduction to Algorithms (6-046J), MIT

https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-046j-introduction-to-algorithms-sma-5503-fall-2005/

και

Lecture Slides for Algorithm Design

https://www.cs.princeton.edu/~wayne/kleinberg-tardos/

Ελάχιστα Γεννητικά Δένδρα

Είσοδος: Ένα συνεκτικό, μη κατευθυνόμενο γράφημα $G = (V, E)$ με συνάρτηση βάρους $w : E \rightarrow \mathrm{R}$.

- Για απλότητα, υποθέτουμε ότι όλα τα βάρη των ακμών είναι διαφορετικά.

Ελάχιστα Γεννητικά Δένδρα

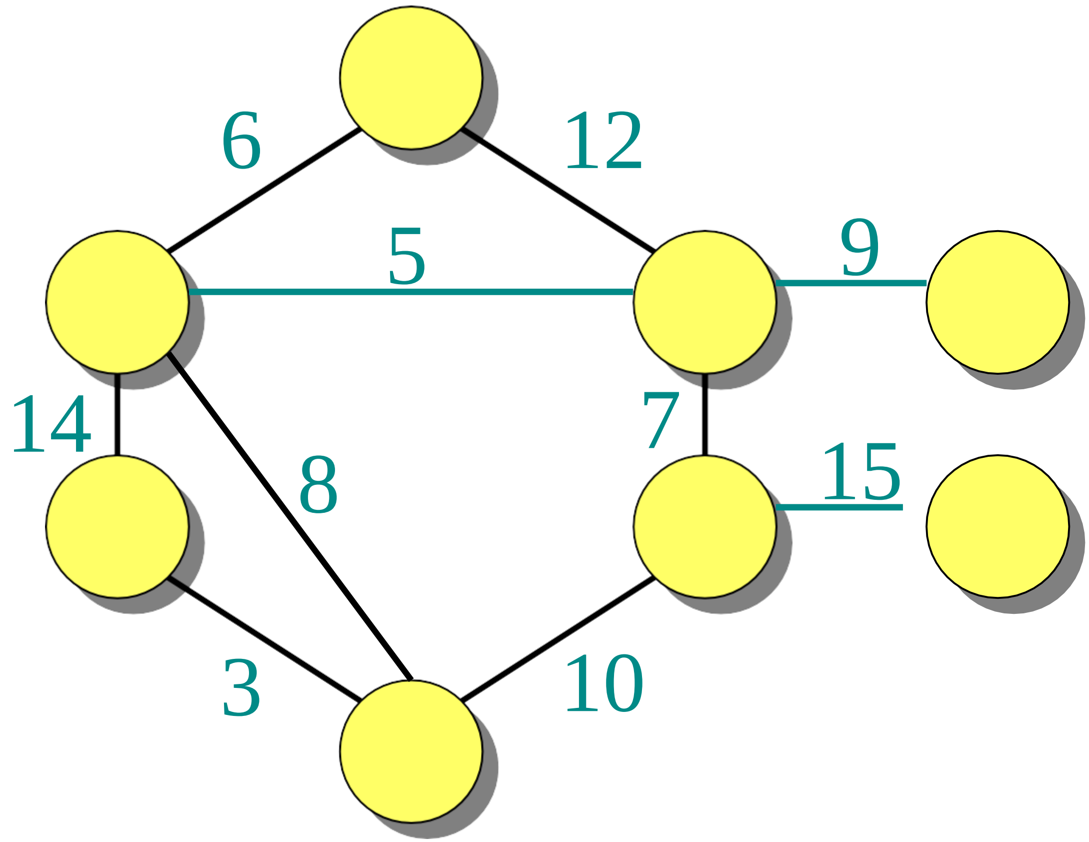
Είσοδος: Ένα συνεκτικό, μη κατευθυνόμενο γράφημα $G = (V, E)$ με συνάρτηση βάρους $w : E \to \mathrm{R}$.

- Για απλότητα, υποθέτουμε ότι όλα τα βάρη των ακμών είναι διαφορετικά

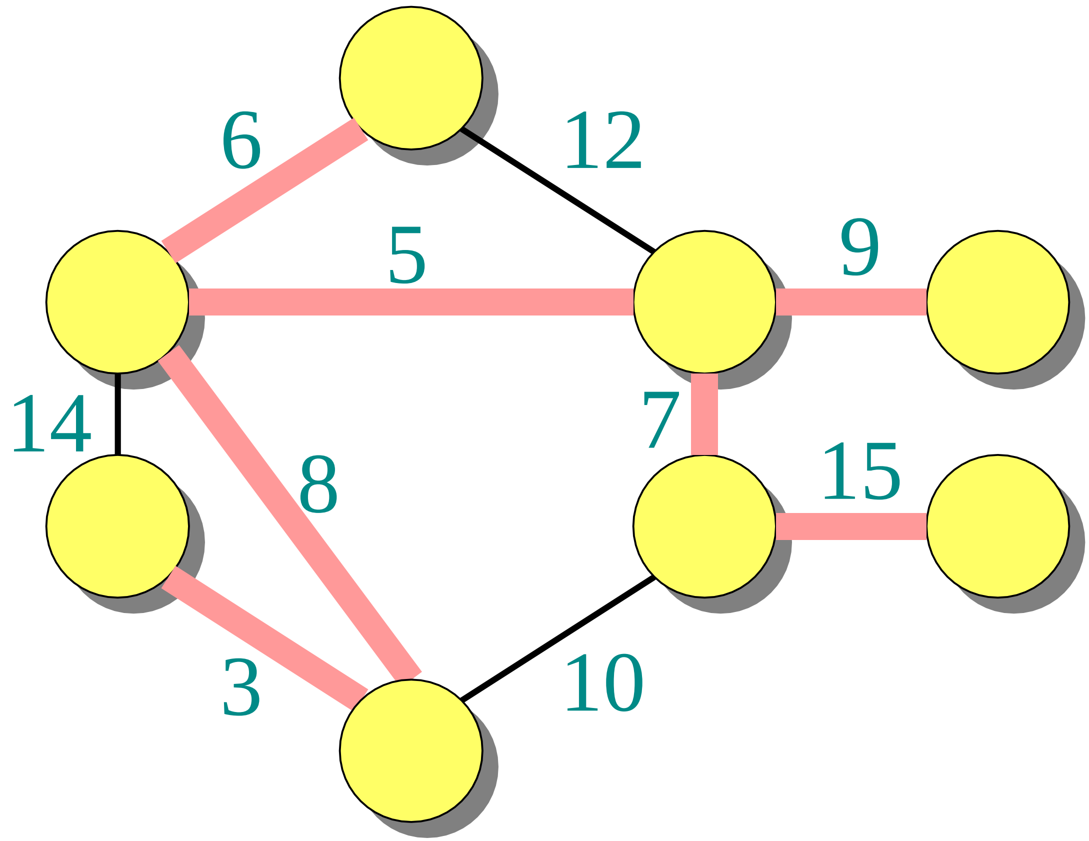
Έξοδος: Ένα ***γεννητικό δέντρο*** T — ένα δέντρο το οποίο περιέχει όλους τους κόμβους — ελάχιστου συνολικά βάρους:

$$w(T) = \sum_{(u,v)\in T} w(u,v).$$

Παράδειγμα ΕΓΔ



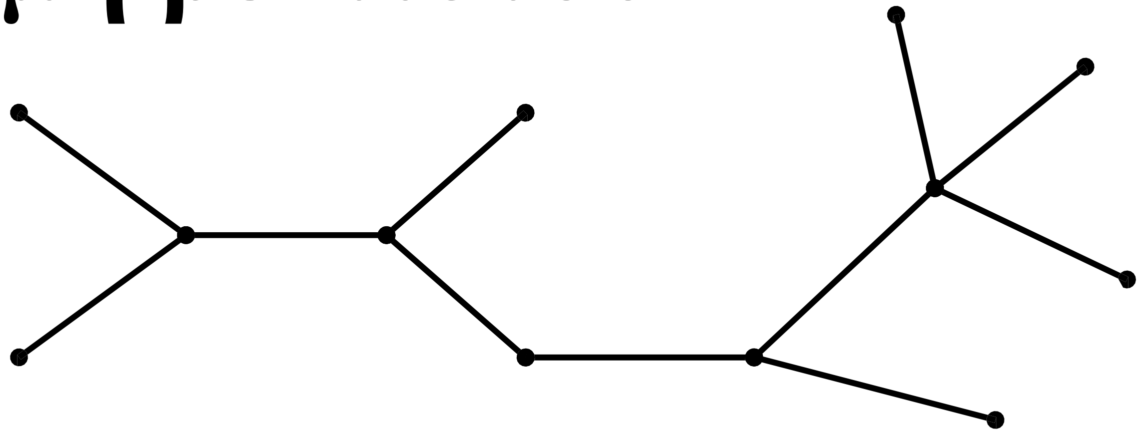
Παράδειγμα ΕΓΔ



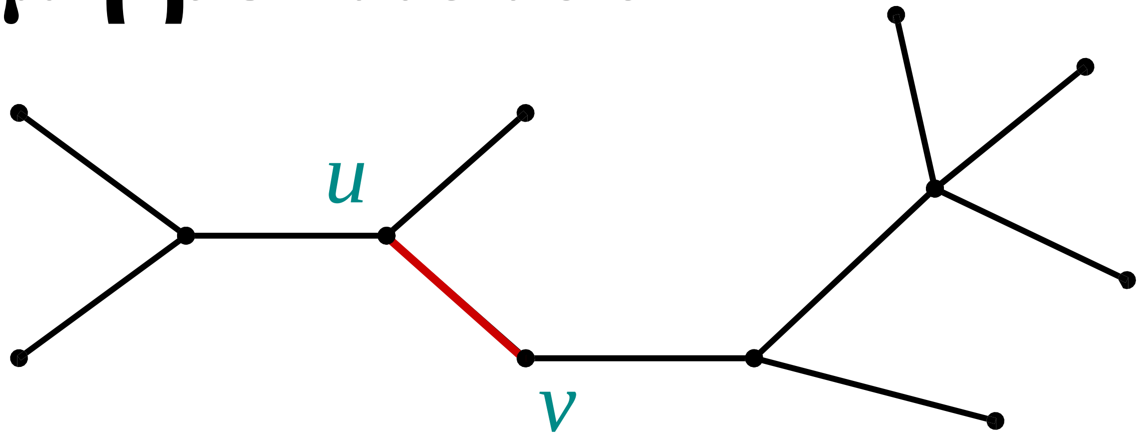
Υποδομή βέλτιστου

ΕΓΔ *T*:

(Οι άλλες ακμές του *G* δεν εμφανίζονται.)

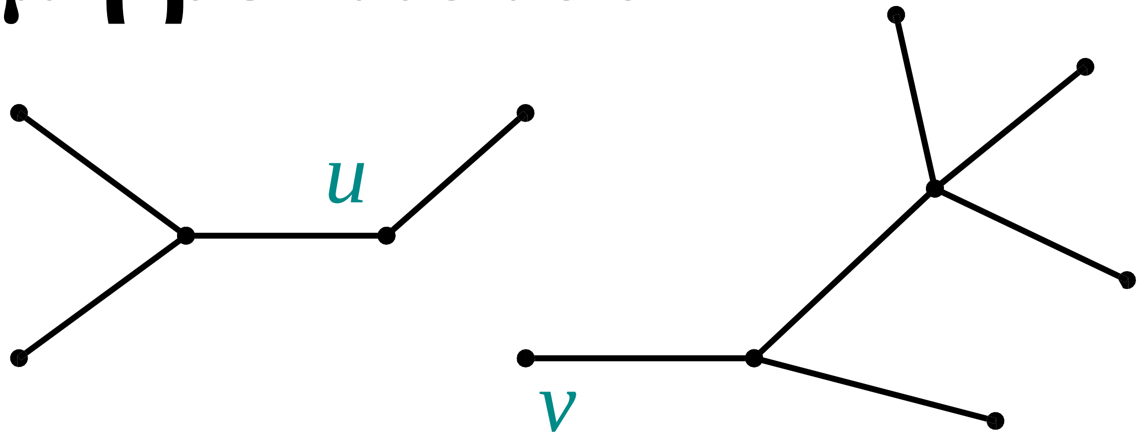


Υποδομή βέλτιστου

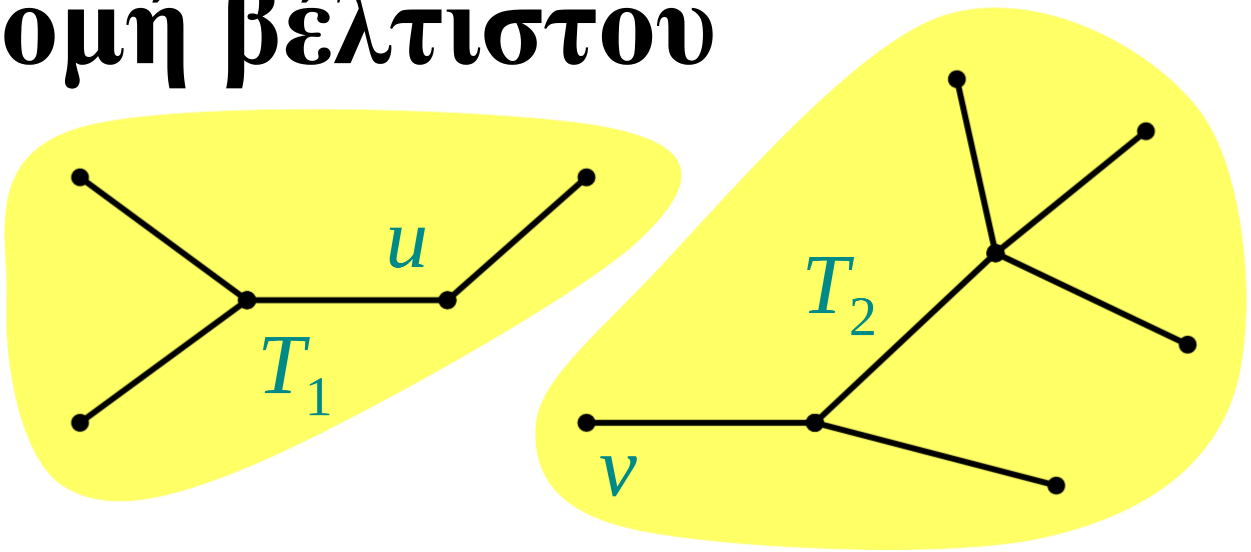


Αφαίρεσε μία ακμή $(u, v) \in T$.

Υποδομή βέλτιστου

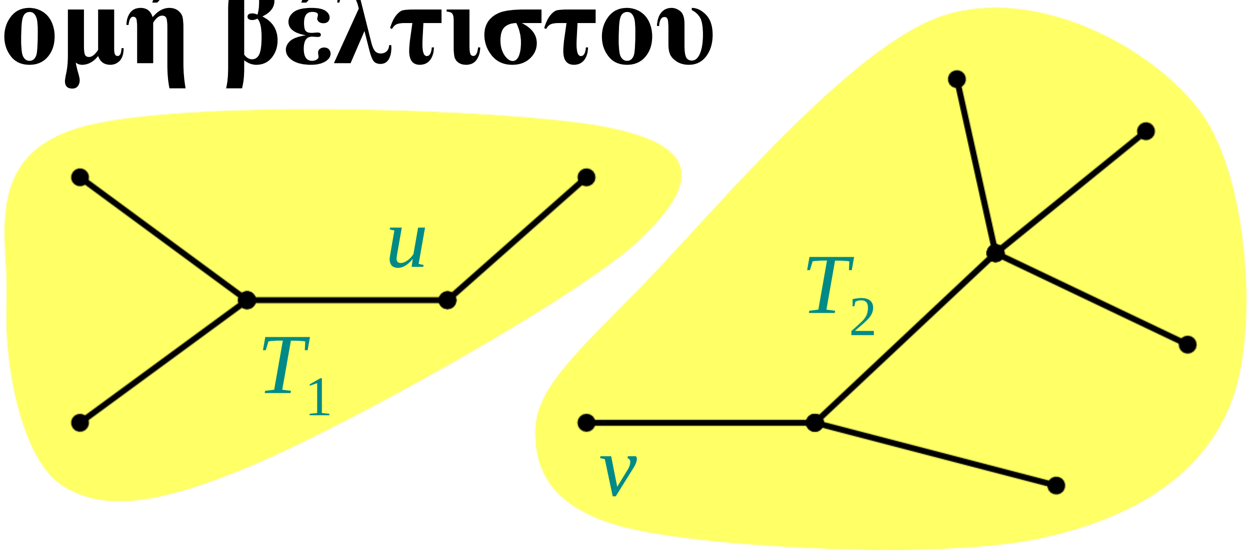


Υποδομή βέλτιστου



Το T χωρίζεται σε δύο υποδέντρα T_1 και T_2.

Υποδομή βέλτιστου



Θεώρημα. Το υποδέντρο T_1 είναι ένα ΕΓΔ του $G_1 = (V_1, E_1)$, το υπογράφημα που επάγεται από τις κορυφές του T_1:

$$V_1 = \text{κορυφές του } T_1,$$
$$E_1 = \{(x, y) \in E : x, y \in V_1\}.$$

Ομοίως για T_2.

Απόδειξη της υποδομής βέλτιστου

Απόδειξη. Αποκοπή και επικόλληση:

$$w(T) = w(u, v) + w(T_1) + w(T_2).$$

Αν T_1' ήταν ένα χαμηλότερου βάρους γεννητικό δέντρο από ότι το T_1 για το G_1, τότε $T' = \{(u, v)\} \cup T_1' \cup T_2$ θα ήταν ένα χαμηλότερου βάρους γεννητικό δέντρο από ότι το T για το G.

Απόδειξη της υποδομής του βέλτιστου

Απόδειξη. Αποκοπή και επικόλληση:

$$w(T) = w(u, v) + w(T_1) + w(T_2).$$

Αν T_1' ήταν ένα χαμηλότερου βάρους γεννητικό δέντρο από ότι το T_1 για το G_1, τότε $T' = \{(u, v)\} \cup T_1' \cup T_2$ θα ήταν ένα χαμηλότερου βάρους γεννητικό δέντρο από ότι το T για το G.

Υπάρχουν επικαλυπτόμενα προβλήματα;

- Ναι.

Απόδειξη της υποδομής του βέλτιστου

Απόδειξη. Αποκοπή και επικόλληση:

$$w(T) = w(u, v) + w(T_1) + w(T_2).$$

Αν T_1' ήταν ένα χαμηλότερου βάρους γεννητικό δέντρο από ότι το T_1 για το G_1, τότε $T' = \{(u, v)\} \cup T_1' \cup T_2$ θα ήταν ένα χαμηλότερου βάρους γεννητικό δέντρο από ότι το T για το G.

Υπάρχουν επικαλυπτόμενα προβλήματα;

- Ναι.

Ο δυναμικός προγραμματισμός μπορεί να εφαρμοσθεί.

- Ναι, αλλά το πρόβλημα του ΕΓΔ έχει άλλη μία ισχυρή ιδιότητα που οδηγεί σε ακόμα πιο αποδοτικό αλγόριθμο.

Το χαρακτηριστικό των “άπληστων” αλγορίθμων

Η ιδιότητα της άπληστης επιλογής

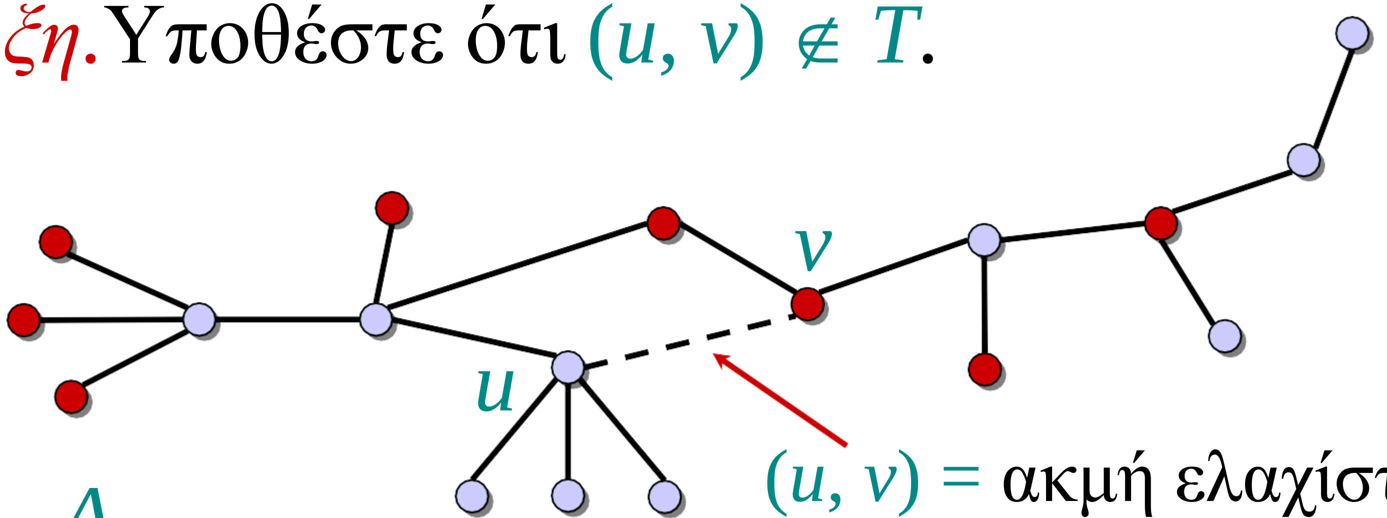
Μία τοπικά βέλτιστη επιλογή είναι και συνολικά βέλτιστη.

Θεώρημα. Έστω T είναι το ΕΓΔ του $G = (V, E)$, και έστω $A \subseteq V$. Υποθέσετε ότι $(u, v) \in E$ είναι η ακμή ελαχίστου βάρους που συνδέει το A με το $V - A$. Τότε, $(u, v) \in T$.

Απόδειξη του Θεωρήματος

Απόδειξη. Υποθέστε ότι $(u, v) \notin T$.

T:



$\in A$

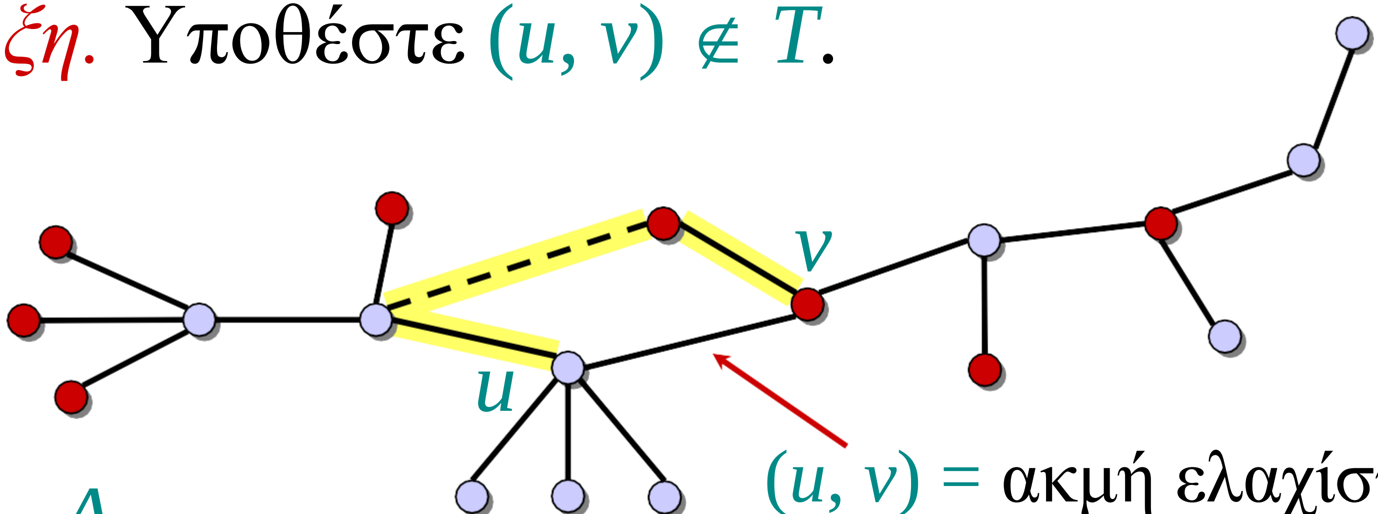
$\in V - A$

(u, v) = ακμή ελαχίστου βάρους που συνδέει A με $V - A$

Απόδειξη Θεωρήματος

Απόδειξη. Υποθέστε $(u, v) \notin T$.

T:



$\in A$

$\in V - A$

$(u, v) =$ ακμή ελαχίστου βάρους που συνδέει το A με το $V - A$

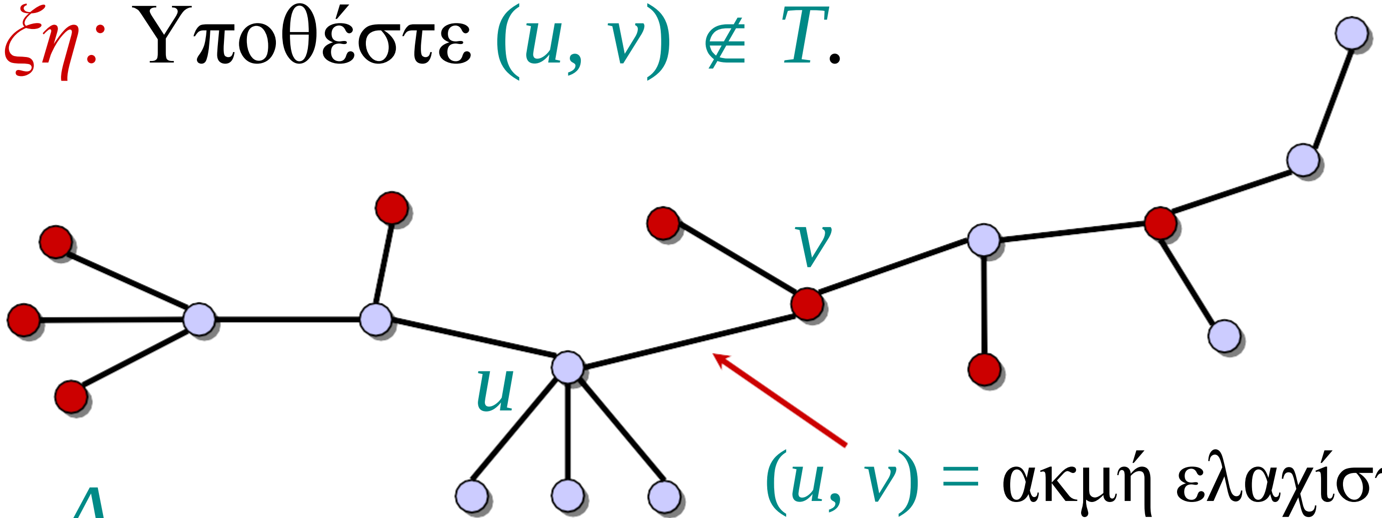
Θεωρήστε το μοναδικό απλό μονοπάτι από το u στο v εντός T.

Ενάλλαξε τη (u, v) με τη πρώτη ακμή στο μονοπάτι η οποία συνδέει μία κορυφή από το A με μία κορυφή στο $V - A$.

Απόδειξη Θεωρήματος

Απόδειξη: Υποθέστε $(u, v) \notin T$.

T':



$\in A$

$\in V - A$

(u, v) = ακμή ελαχίστου βάρους που συνδέει το A με το $V - A$

Προκύπτει ένα γεννητικό δέντρο χαμηλότερου συνολικά βάρους από ότι το T.

Αλγόριθμος του Prim

Ιδέα: Αποθήκευσε τους κόμβους του $V - A$ σε μία ουρά προτεραιότητας Q. Το κλειδί κάθε κόμβου στο Q είναι το βάρος της ακμής ελαχίστου βάρους που συνδέει τον κόμβο αυτό με ένα κόμβο στο A.

$Q \leftarrow V$

$key[v] \leftarrow \infty$ for all $v \in V$

$key[s] \leftarrow 0$ for some arbitrary $s \in V$

while $Q \neq \emptyset$

  do $u \leftarrow$ EXTRACT-MIN(Q)

   for each $v \in Adj[u]$

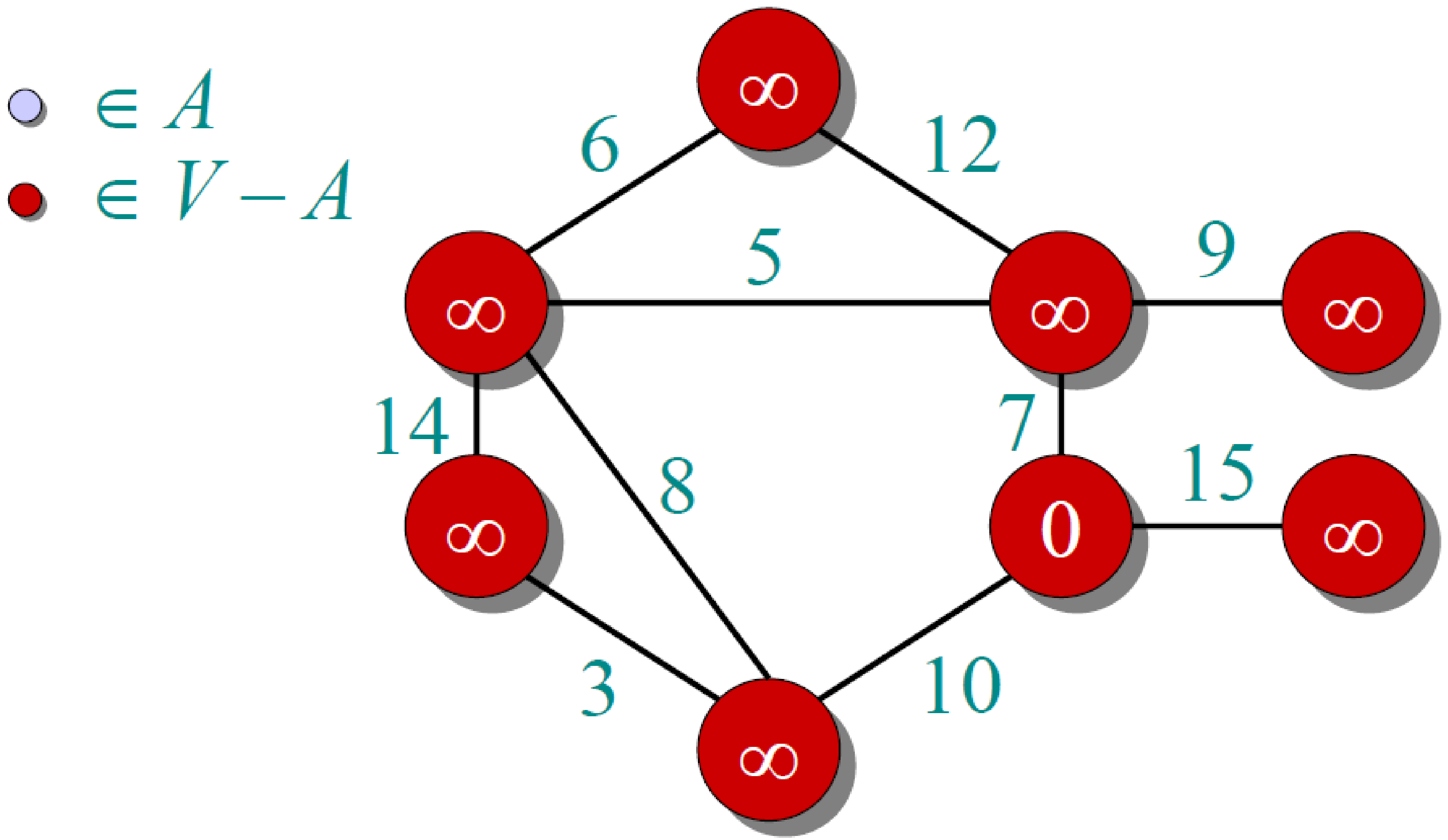
    do if $v \in Q$ and $w(u, v) < key[v]$ ⊳ DECREASE-KEY

     then $key[v] \leftarrow w(u, v)$

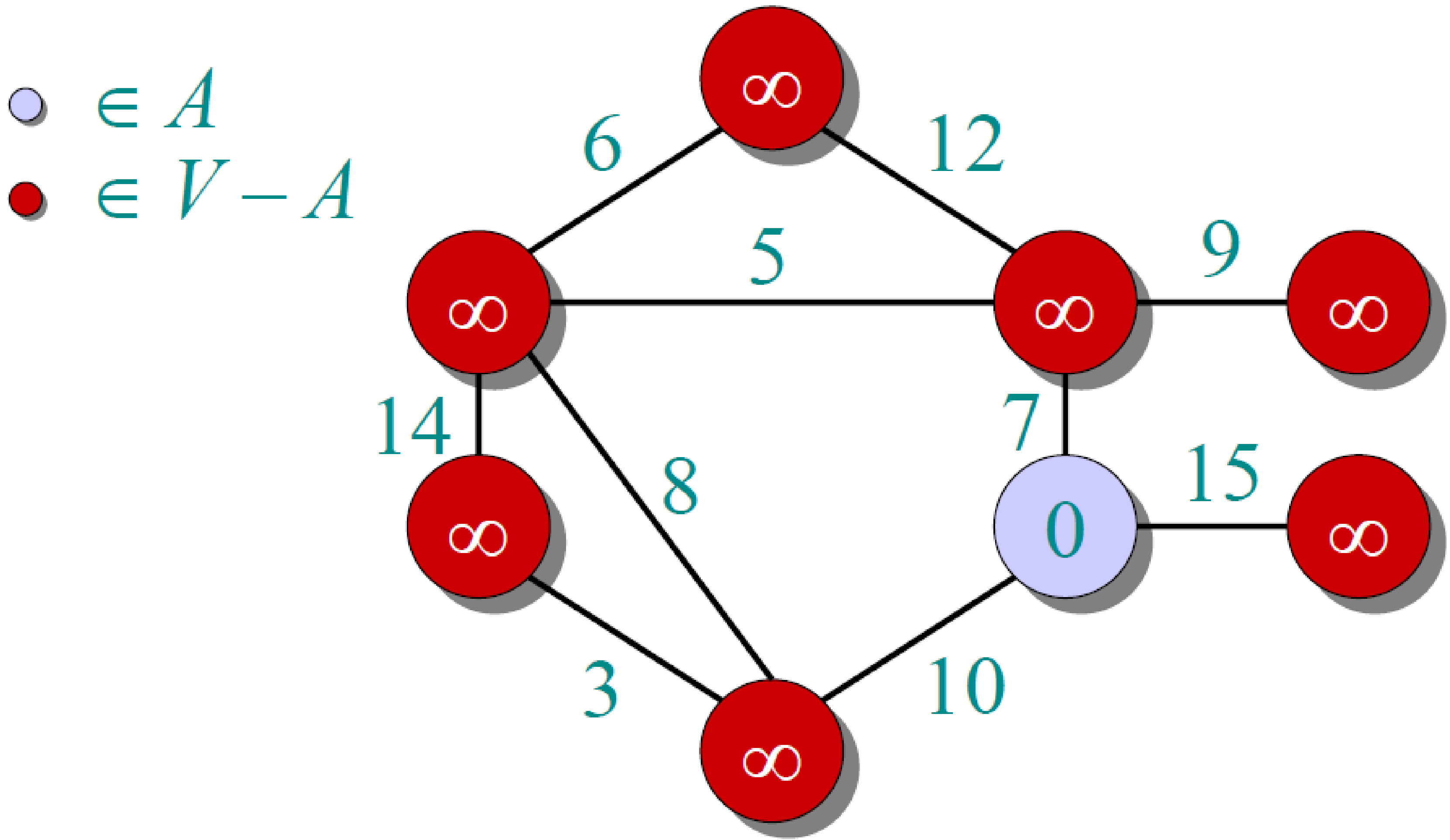
       $\pi[v] \leftarrow u$

Στο τέλος, $\{(v, \pi[v])\}$ σχηματίζουν το MST.

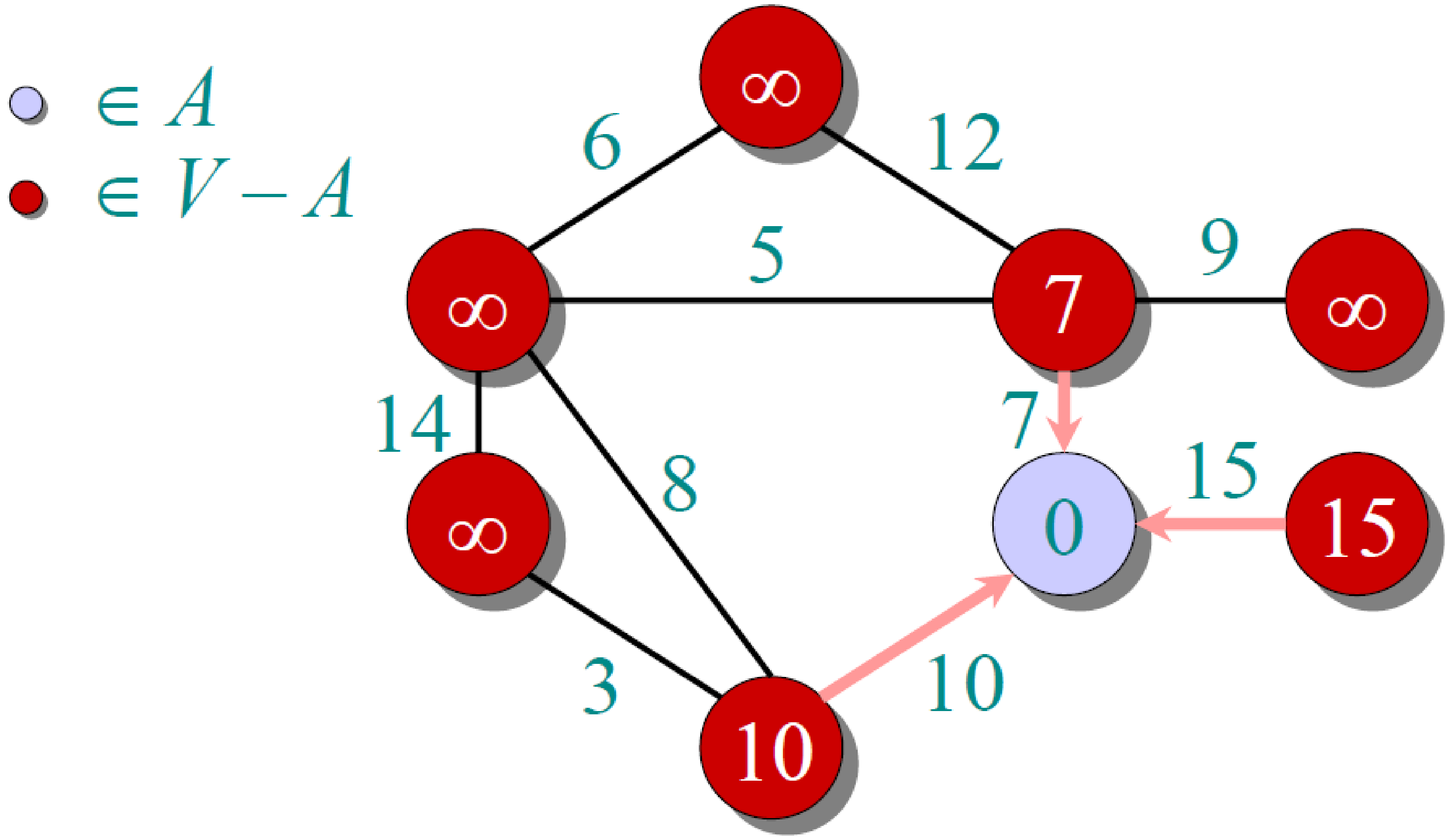
Παράδειγμα αλγορίθμου Prim



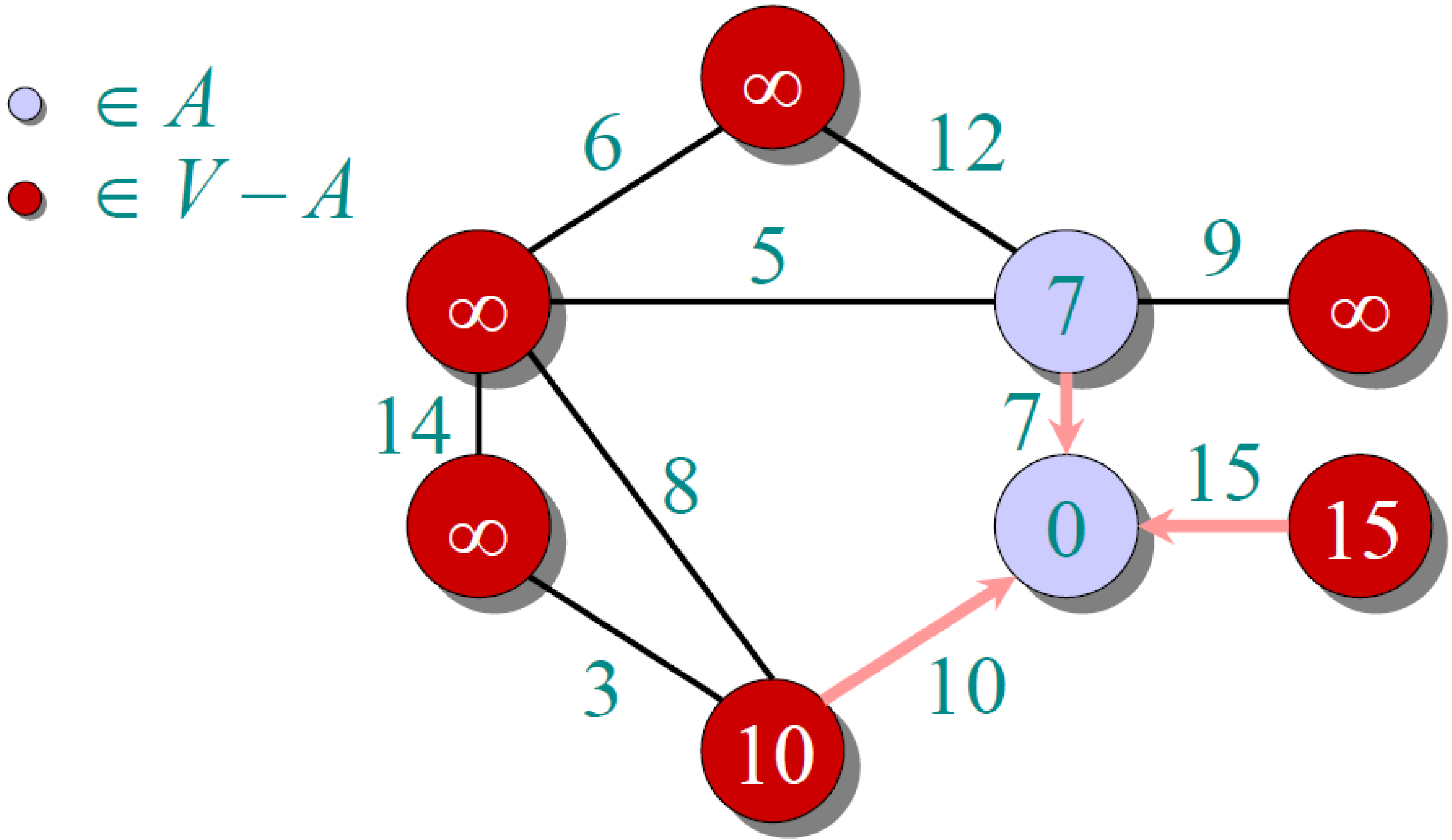
Παράδειγμα αλγορίθμου Prim



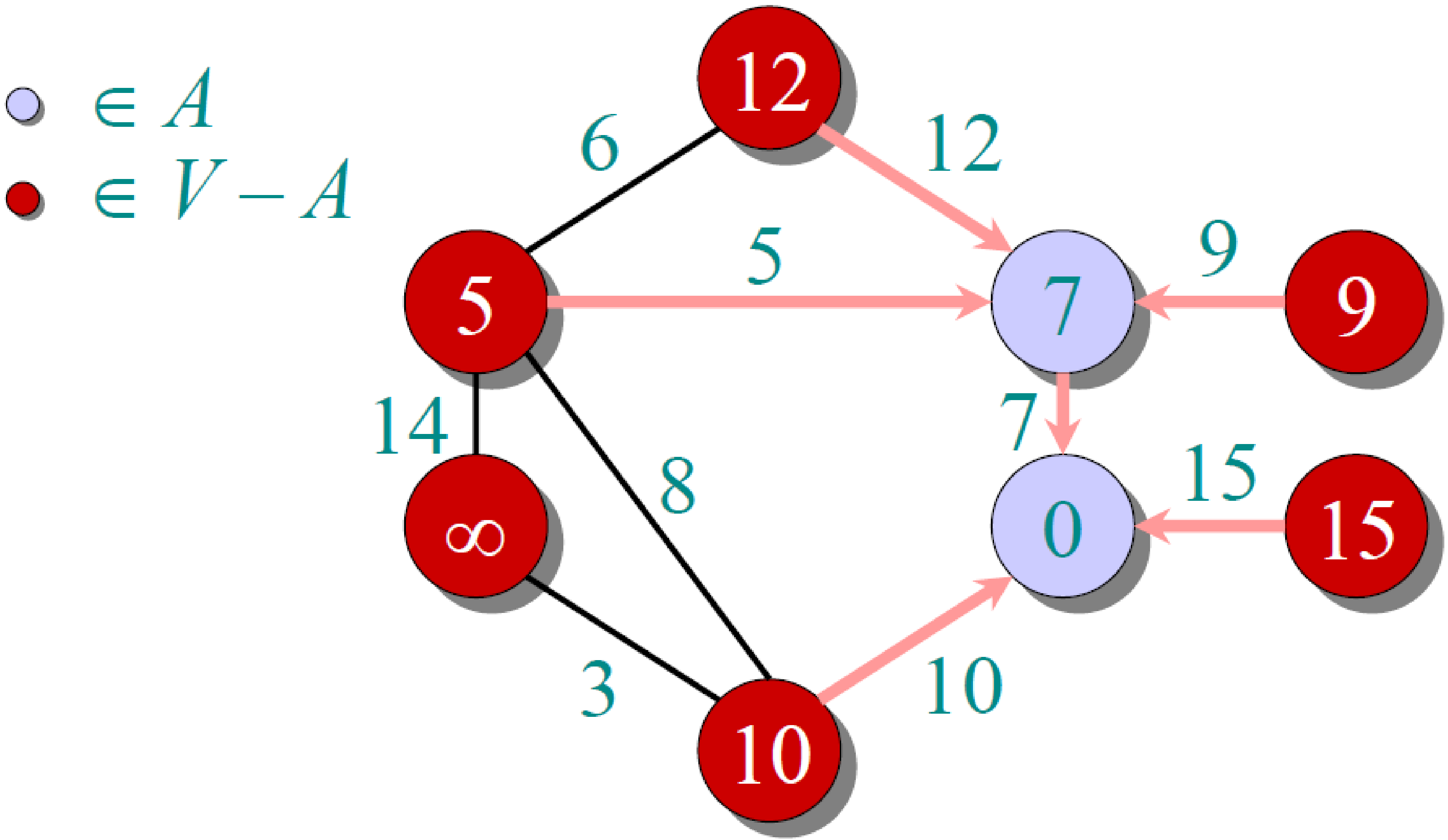
Παράδειγμα αλγορίθμου Prim



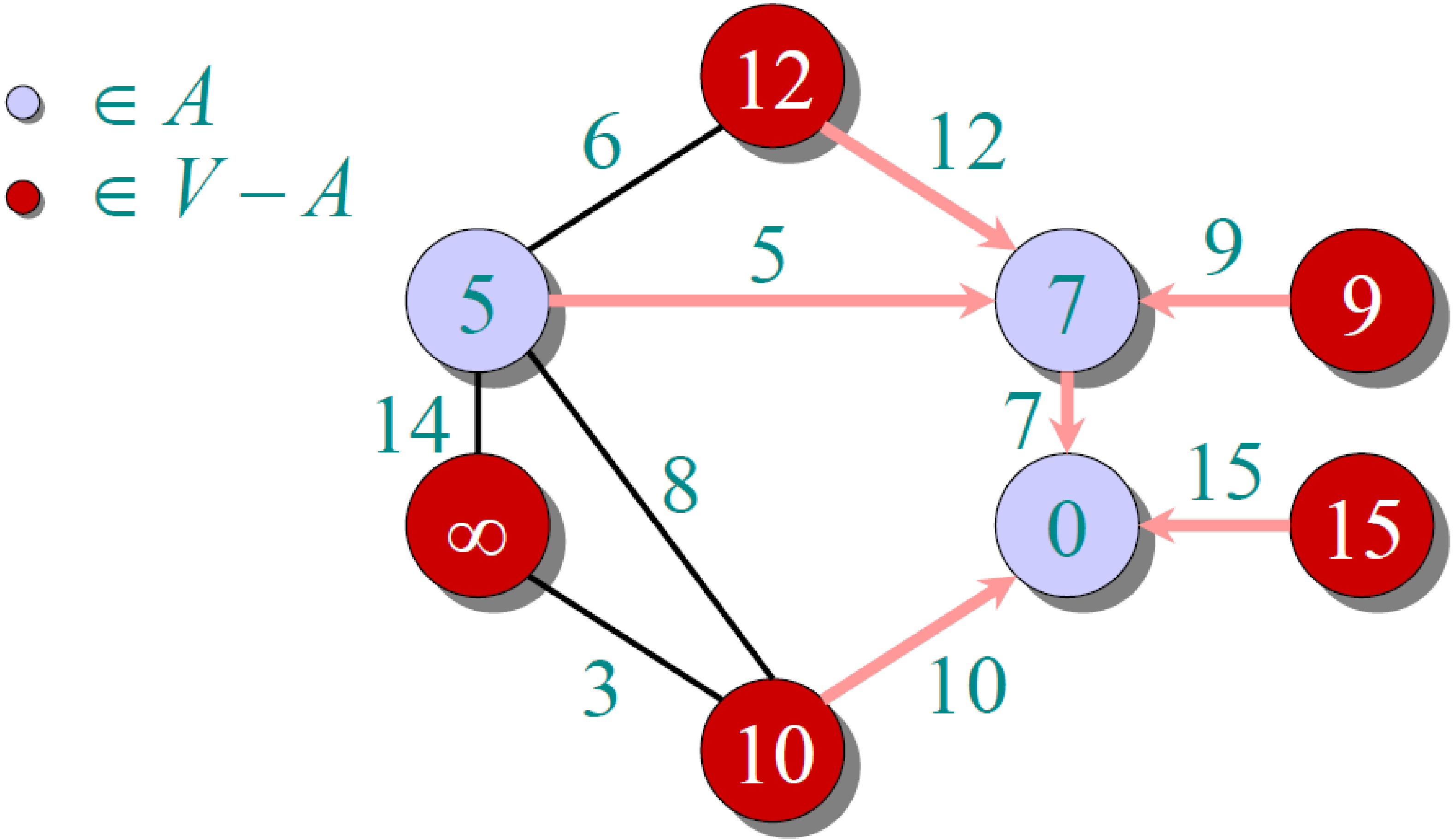
Παράδειγμα αλγορίθμου Prim



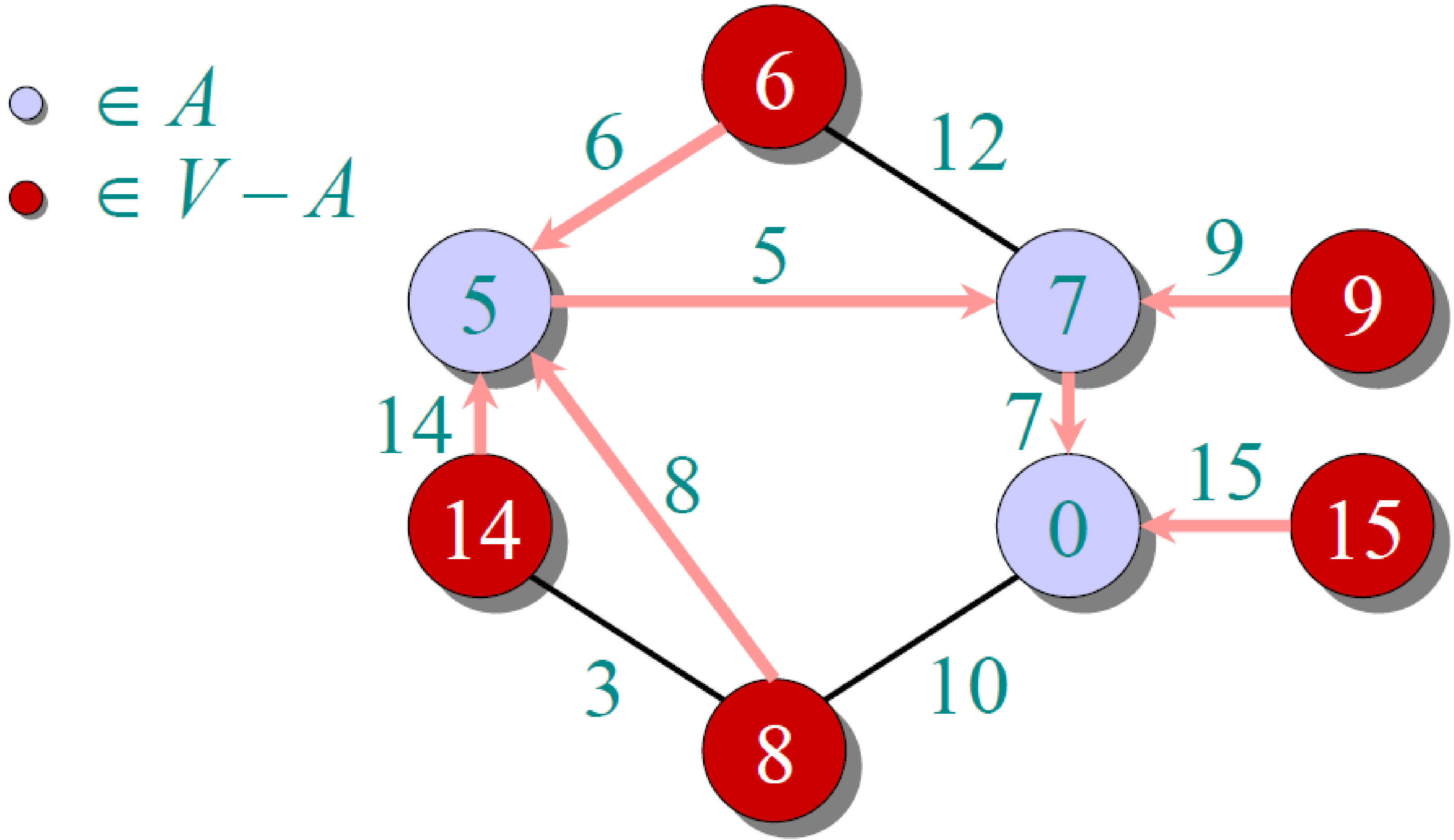
Παράδειγμα αλγορίθμου Prim



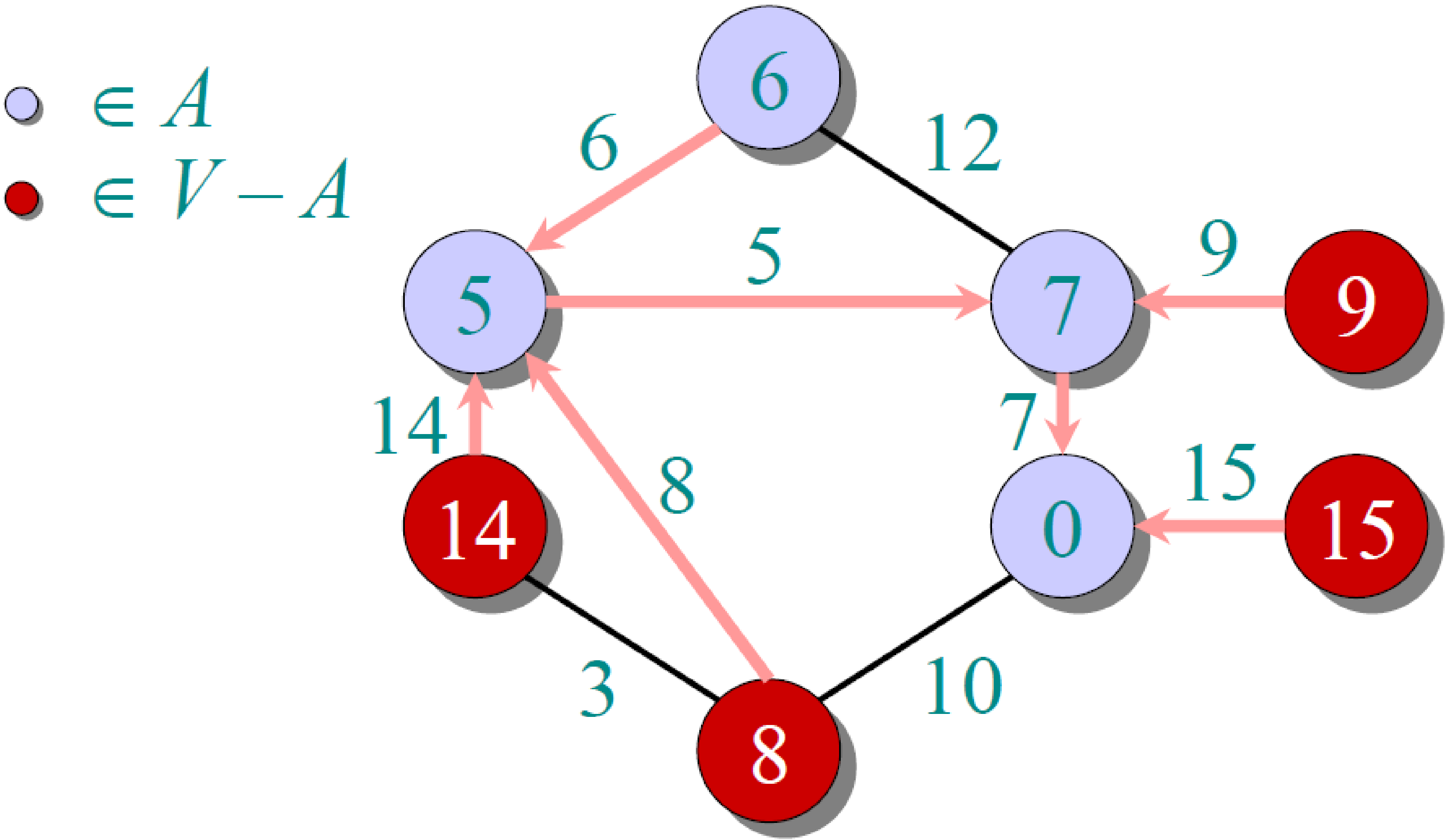
Παράδειγμα αλγορίθμου Prim



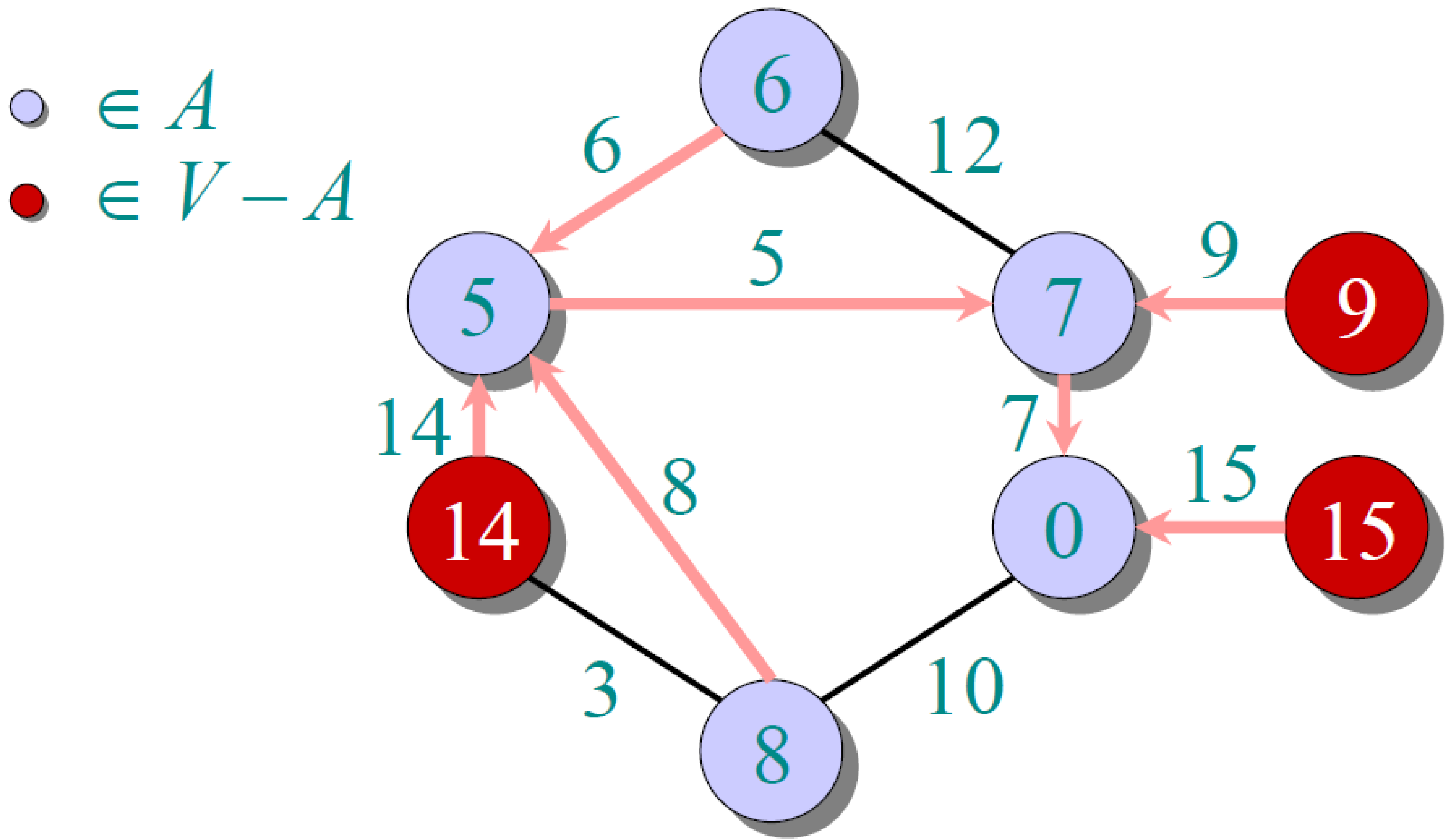
Παράδειγμα αλγορίθμου Prim



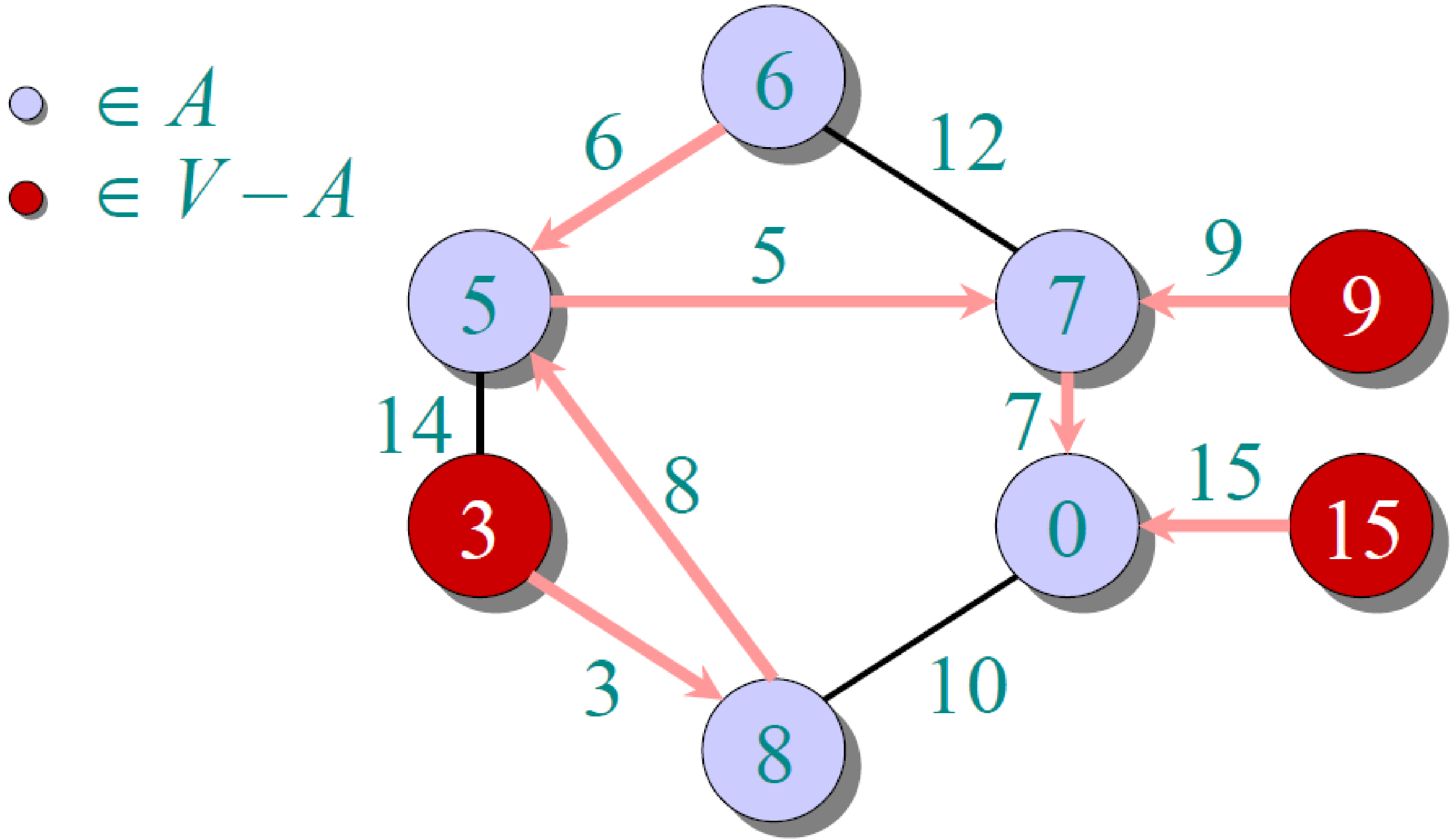
Παράδειγμα αλγορίθμου Prim



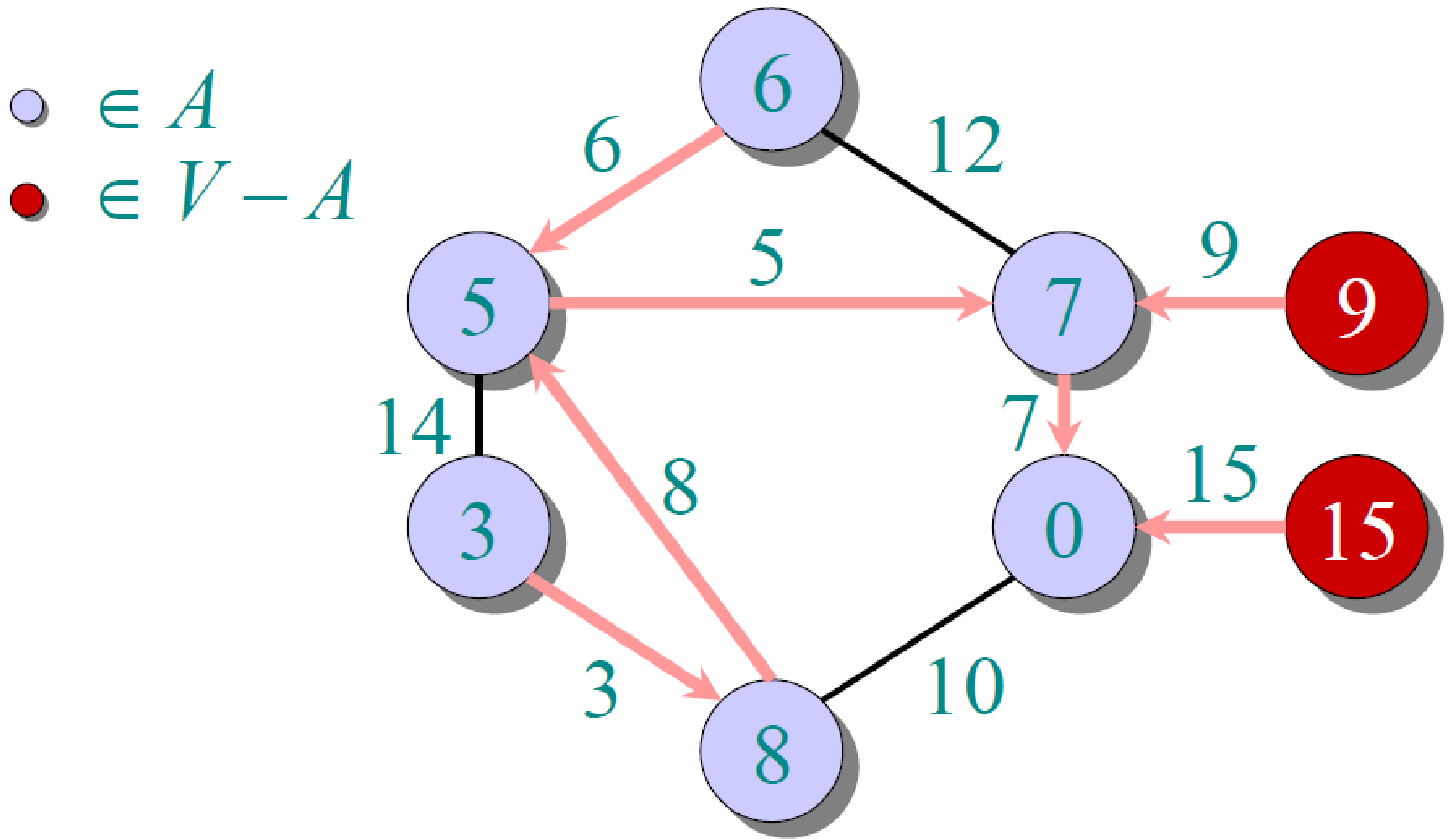
Παράδειγμα αλγορίθμου Prim



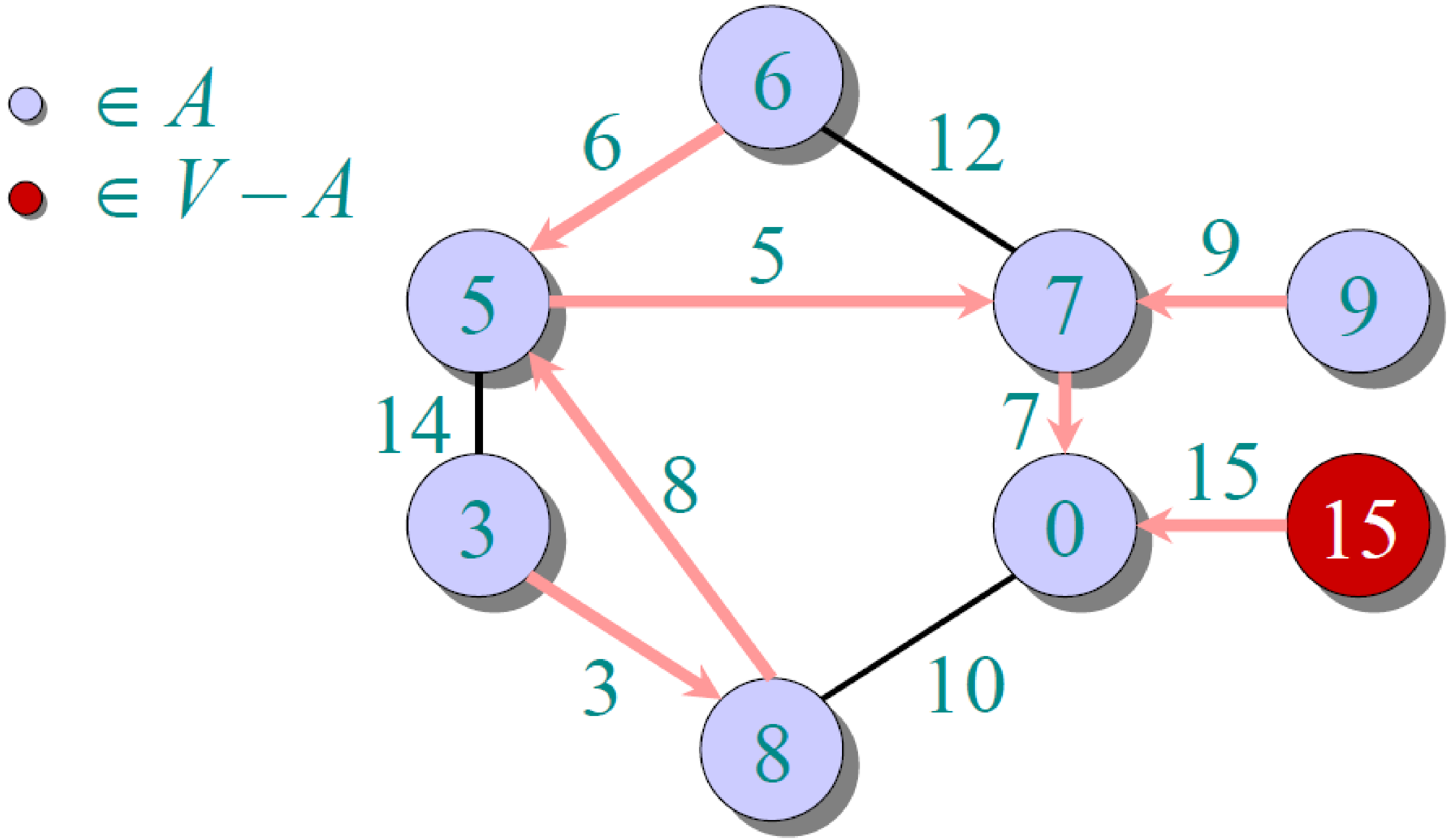
Παράδειγμα αλγορίθμου Prim



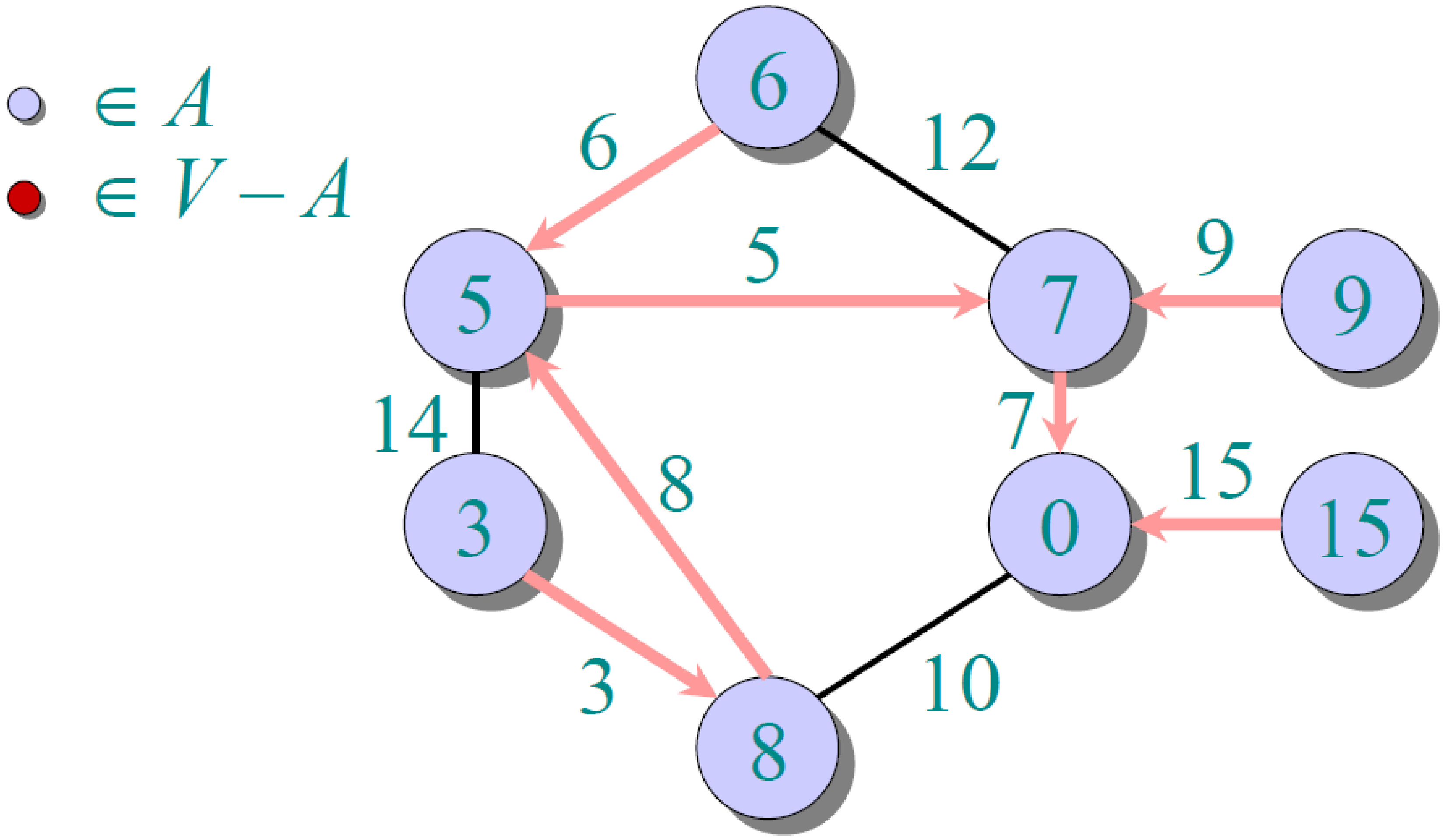
Παράδειγμα αλγορίθμου Prim

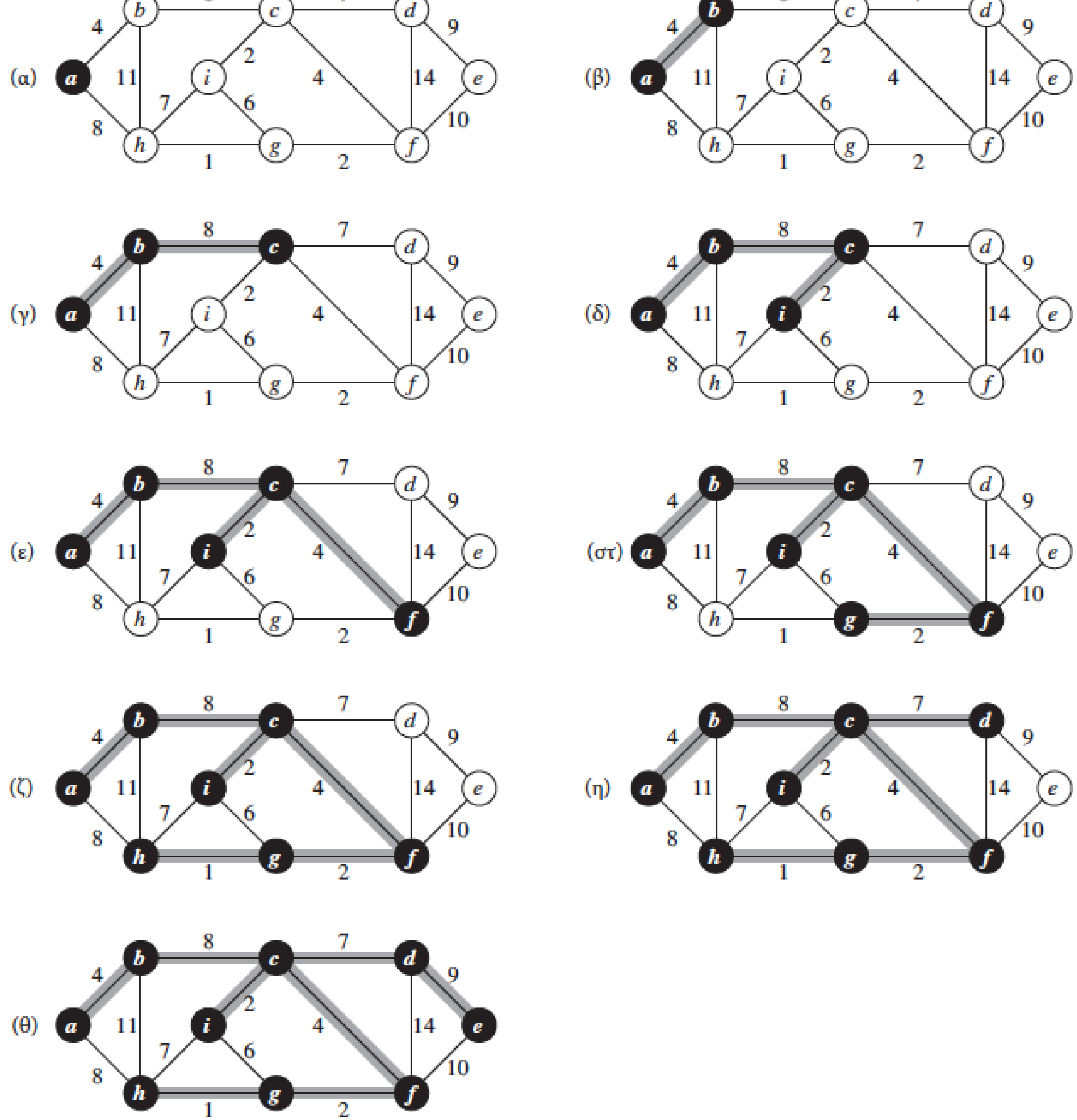


Παράδειγμα αλγορίθμου Prim



Παράδειγμα αλγορίθμου Prim





Σχήμα 23.5 Η λειτουργία του αλγορίθμου του Prim στο γράφημα του Σχήματος 23.1. Ο ριζικός κόμβος είναι ο a. Οι σκιασμένες ακμές και οι μαύροι κόμβοι ανήκουν στο αναπτυσσόμενο δένδρο. Σε κάθε βήμα του αλγορίθμου, οι κόμβοι του δένδρου ορίζουν μια τομή του γραφήματος, και η ακμή που προστίθεται στο δένδρο είναι μια ελαφρά ακμή ως προς τη διάσχιση αυτής της τομής. Στο δεύτερο βήμα, παραδείγματος χάριν, ο αλγόριθμος έχει τη δυνατότητα να προσθέσει στο δένδρο είτε την ακμή (b, c) είτε την ακμή (a, h), αφού και οι δύο είναι ελαφρείς ως προς τη διάσχιση της τομής.

Ανάλυση του αλγορίθμου του Prim

$Q \leftarrow V$
$key[v] \leftarrow \infty$ for all $v \in V$
$key[s] \leftarrow 0$ for some arbitrary $s \in V$
while $Q \neq \emptyset$
  do $u \leftarrow$ EXTRACT-MIN(Q)
    for each $v \in Adj[u]$
      do if $v \in Q$ and $w(u, v) < key[v]$
        then $key[v] \leftarrow w(u, v)$
          $\pi[v] \leftarrow u$

Ανάλυση του αλγορίθμου του Prim

$$\begin{array}{c}\Theta(V)\\ \text{συνολικά}\end{array} \left\{ \begin{array}{l} Q \leftarrow V \\ key[v] \leftarrow \infty \text{ for all } v \in V \\ key[s] \leftarrow 0 \text{ for some arbitrary } s \in V \end{array} \right.$$

while $Q \neq \emptyset$

  do $u \leftarrow$ EXTRACT-MIN(Q)

    for each $v \in Adj[u]$

      do if $v \in Q$ and $w(u, v) < key[v]$

        then $key[v] \leftarrow w(u, v)$

          $\pi[v] \leftarrow u$

Ανάλυση του αλγορίθμου του Prim

$\Theta(V)$ συνολικά

$Q \leftarrow V$

$key[v] \leftarrow \infty$ for all $v \in V$

$key[s] \leftarrow 0$ for some arbitrary $s \in V$

$|V|$ φορές

while $Q \neq \emptyset$

    do $u \leftarrow$ EXTRACT-MIN(Q)

        for each $v \in Adj[u]$

            do if $v \in Q$ and $w(u, v) < key[v]$

                then $key[v] \leftarrow w(u, v)$

                    $\pi[v] \leftarrow u$

Ανάλυση του αλγορίθμου του Prim

$\Theta(V)$ συνολικά

$Q \leftarrow V$
$key[v] \leftarrow \infty$ for all $v \in V$
$key[s] \leftarrow 0$ for some arbitrary $s \in V$

$|V|$ φορές

while $Q \neq \emptyset$
  do $u \leftarrow$ EXTRACT-MIN(Q)

$degree(u)$ φορές

    for each $v \in Adj[u]$
      do if $v \in Q$ and $w(u, v) < key[v]$
        then $key[v] \leftarrow w(u, v)$
          $\pi[v] \leftarrow u$

Ανάλυση του αλγορίθμου του Prim

$\Theta(V)$ total

$Q \leftarrow V$
$key[v] \leftarrow \infty$ for all $v \in V$
$key[s] \leftarrow 0$ for some arbitrary $s \in V$

$|V|$ times

while $Q \neq \emptyset$
    do $u \leftarrow$ EXTRACT-MIN(Q)

$degree(u)$ times

        for each $v \in Adj[u]$
            do if $v \in Q$ and $w(u, v) < key[v]$
                then $key[v] \leftarrow w(u, v)$
                      $\pi[v] \leftarrow u$

$\Theta(E)$ DECREASE-KEY's.

Ανάλυση του αλγορίθμου του Prim

$\Theta(V)$ total

$Q \leftarrow V$

$key[v] \leftarrow \infty$ for all $v \in V$

$key[s] \leftarrow 0$ for some arbitrary $s \in V$

$|V|$ times

while $Q \neq \varnothing$

do $u \leftarrow$ EXTRACT-MIN(Q)

$degree(u)$ times

for each $v \in Adj[u]$

do if $v \in Q$ and $w(u, v) < key[v]$

then $key[v] \leftarrow w(u, v)$

$\pi[v] \leftarrow u$

Χρόνος $= \Theta(V) \cdot T_{\text{EXTRACT-MIN}} + \Theta(E) \cdot T_{\text{DECREASE-KEY}}$

Ανάλυση του αλγόριθμου του Prim

Time $= \Theta(V) \cdot T_{\text{EXTRACT-MIN}} + \Theta(E) \cdot T_{\text{DECREASE-KEY}}$

Q	$T_{\text{EXTRACT-MIN}}$	$T_{\text{DECREASE-KEY}}$	Σύνολο

Ανάλυση του αλγόριθμου του Prim

Time $= \Theta(V) \cdot T_{\text{EXTRACT-MIN}} + \Theta(E) \cdot T_{\text{DECREASE-KEY}}$

Q	$T_{\text{EXTRACT-MIN}}$	$T_{\text{DECREASE-KEY}}$	Total
array	$O(V)$	$O(1)$	$O(V^2)$

Ανάλυση του αλγόριθμου του Prim

$$\text{Time} = \Theta(V)\cdot T_{\text{EXTRACT-MIN}} + \Theta(E)\cdot T_{\text{DECREASE-KEY}}$$

Q	$T_{\text{EXTRACT-MIN}}$	$T_{\text{DECREASE-KEY}}$	Total
array	$O(V)$	$O(1)$	$O(V^2)$
binary heap	$O(\lg V)$	$O(\lg V)$	$O(E \lg V)$

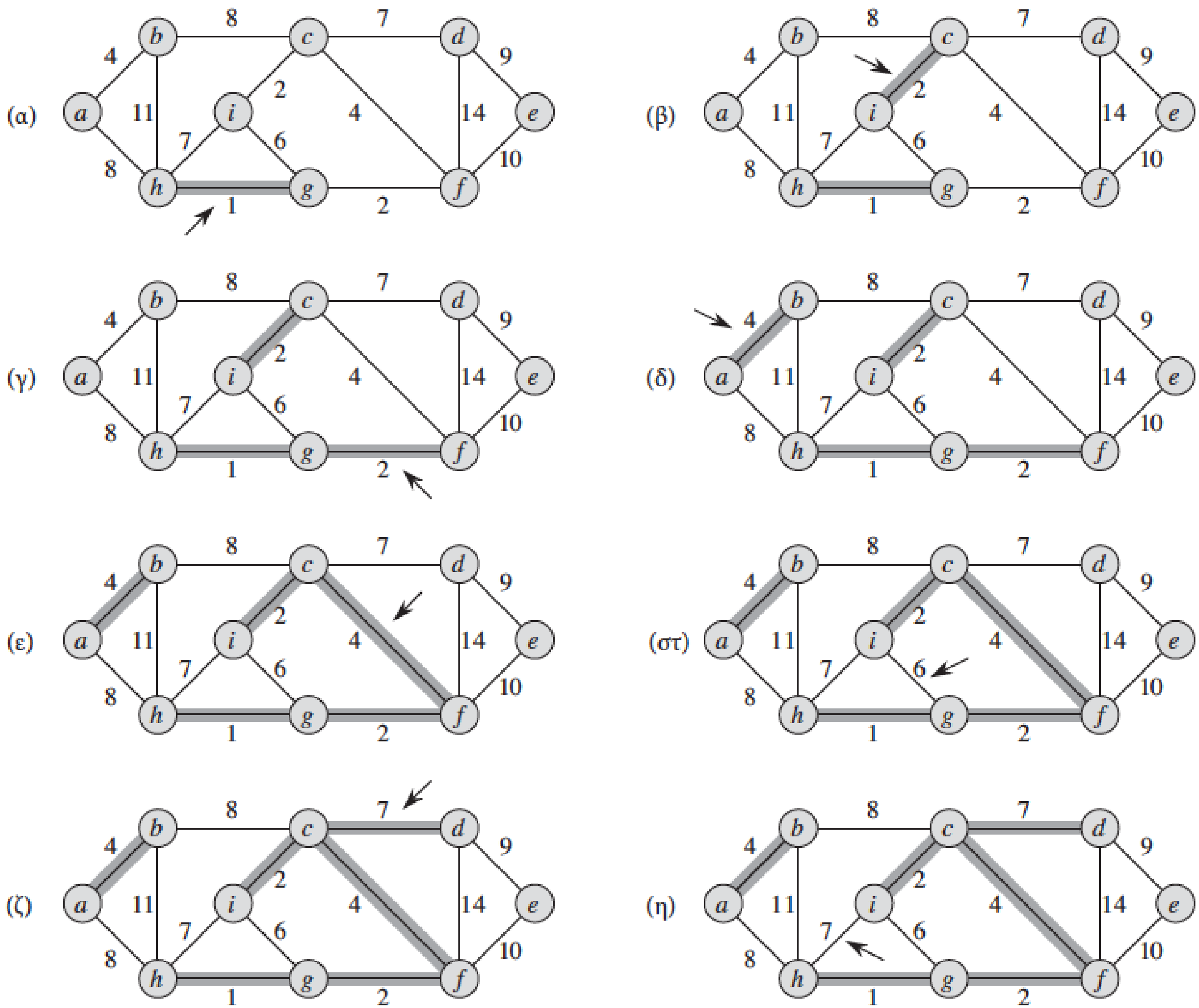
Ανάλυση του αλγόριθμου του Prim

$$\text{Time} = \Theta(V)\cdot T_{\textsc{Extract-Min}} + \Theta(E)\cdot T_{\textsc{Decrease-Key}}$$

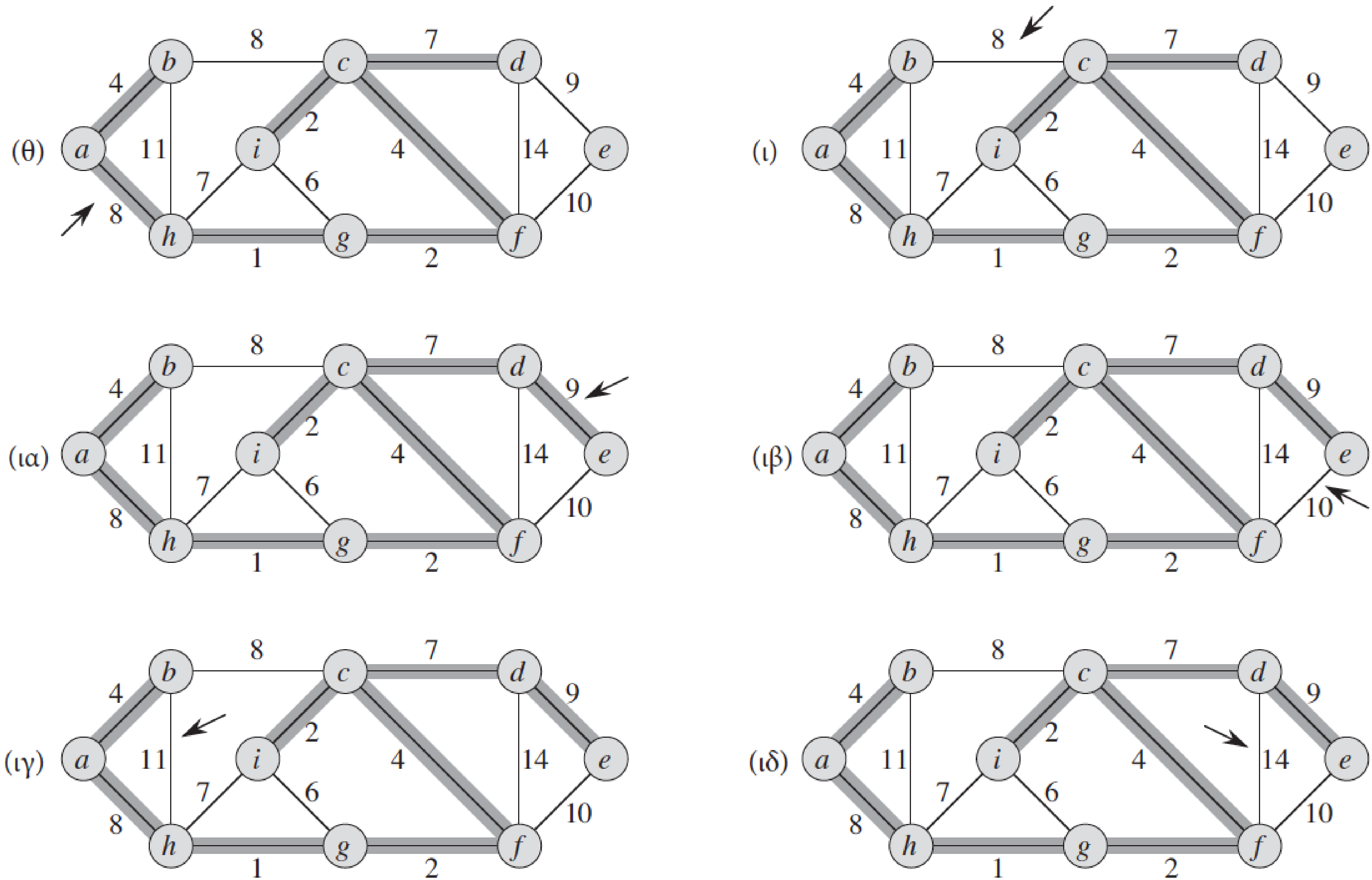
Q	$T_{\textsc{Extract-Min}}$	$T_{\textsc{Decrease-Key}}$	Σύνολο
array	$O(V)$	$O(1)$	$O(V^2)$
binary heap	$O(\lg V)$	$O(\lg V)$	$O(E \lg V)$
Fibonacci heap	$O(\lg V)$ Επιμερισμένο	$O(1)$ Επιμερισμένο	$O(E + V \lg V)$ Χειρότερης Περίπτωσης

Αλγόριθμος του Kruskal

```
MST-KRUSKAL(G, w)
A = ∅
for each vertex v ∈ G.V
    MAKE-SET(v)
sort the edges of G.E into nondecreasing order by weight w
for each edge (u, v) ∈ G.E, taken in nondecreasing order by weight
    if FIND-SET(u) ≠ FIND-SET(v)
        A = A ∪ {(u, v)}
        UNION(u, v)
return A
```



Σχήμα 23.4 Η λειτουργία του αλγορίθμου του Kruskal στο γράφημα του Σχήματος 23.1. Οι σκιασμένες ακμές είναι αυτές που ανήκουν στο αναπτυσσόμενο δάσος A. Ο αλγόριθμος εξετάζει τις ακμές με βάση τη διάταξή τους ως προς το βάρος. Η εξεταζόμενη ακμή σε κάθε βήμα του αλγορίθμου υποδεικνύεται από το βέλος. Εάν η ακμή αυτή συνδέει δύο διαφορετικά δένδρα του δάσους, προστίθεται στο δάσος, συγχωνεύοντας με τον τρόπο αυτό τα δύο δένδρα.



Σχήμα 23.4, συνέχ. Επιπλέον βήματα στην εκτέλεση του αλγορίθμου του Kruskal.

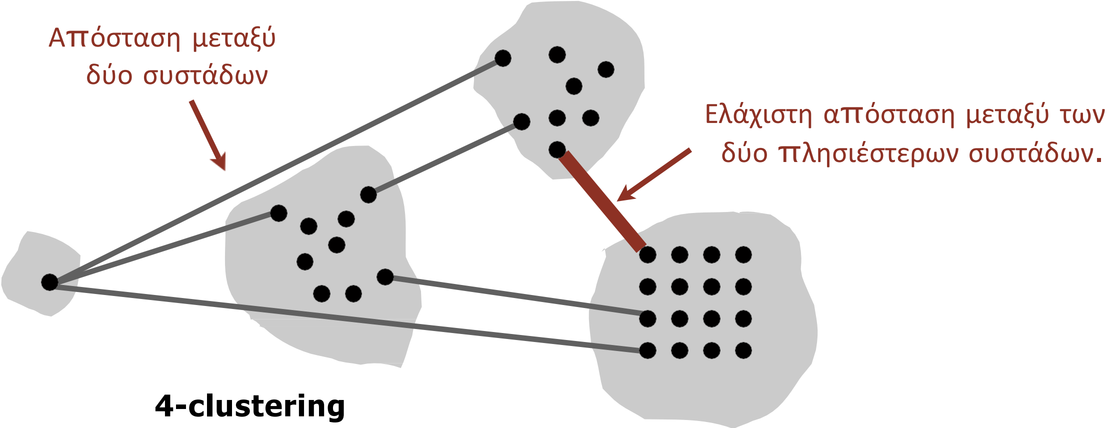
Δημιουργία συστάδων μέγιστου διαχωρισμού (spacing)

k-clustering. Διαίρεσε τα αντικείμενα σε k μη κενές ομάδες.
Συνάρτηση απόστασης. Αριθμητική τιμή η οποία προσδιορίζει την «εγγύτητα» δύο αντικειμένων.

- $d(p_i, p_j) = 0$ iff $p_i = p_j$
- $d(p_i, p_j) \geq 0$
- $d(p_i, p_j) = d(p_j, p_i)$

Διαχωρισμός. Ελάχιστη απόσταση μεταξύ δύο σημείων που ανήκουν σε διαφορετικές συστάδες.

Στόχος. Δοθέντος ενός ακεραίου k, βρες k συστάδες μέγιστου διαχωρισμού.

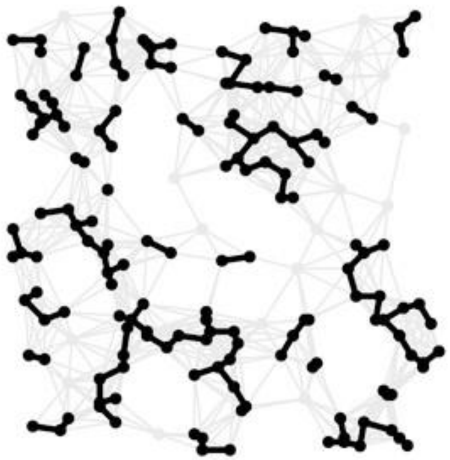


4-clustering

Άπληστος αλγόριθμος δημιουργίας συστάδων

Γνωστός αλγόριθμος στη βιβλιογραφία για single-linkage k-clustering:

- Σχημάτισε ένα γράφημα πάνω στο σύνολο των κόμβων U, που αντιστοιχεί σε n συστάδες.
- Βρες το πλησιέστερο ζεύγος αντικειμένων τέτοιο ώστε κάθε αντικείμενο είναι σε διαφορετική συστάδα, και πρόσθεσε μία ακμή μεταξύ τους.
- Επανέλαβε $n - k$ φορές (μέχρι να υπάρχουν ακριβώς k συστάδες).



Βασική παρατήρηση. Αυτή η διαδικασία είναι ακριβώς ο αλγόριθμος του Kruskal (εκτός του ότι σταματούμε όταν υπάρχουν k συνεκτικές συνιστώσες).

Εναλλακτικά. Βρες ένα ελάχιστο γεννητικό δέντρο και αφαίρεσε τις $k-1$ μακρύτερες ακμές.

Άπληστος αλγόριθμος δημιουργίας συστάδων: ανάλυση

Θεώρημα. Έστω C^* το σύνολο των συστάδων $C_1^*, \ldots, C_k^*$ που σχηματίζονται με τη αφαίρεση των $k-1$ μακρύτερων ακμών σε ένα ελάχιστο γεννητικό δέντρο.

Τότε, C^* είναι ένα σύνολο k συστάδων μέγιστου διαχωρισμού.

Pf. Έστω C οποιαδήποτε άλλη ομάδα k συστάδων $C_1, \ldots, C_k$.

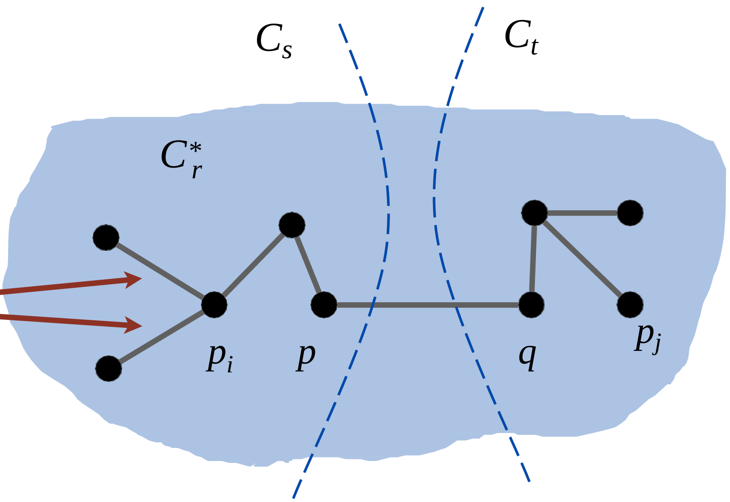
Έστω p_i και p_j είναι στην ίδια συστάδα στο C^*, έστω C_r^*, αλλά σε διαφορετικές συστάδες στη C, έστω C_s και C_t.

Κάποια ακμή (p, q) στο μονοπάτι $p_i - p_j$ στο C_r^* διασχίζει δύο διαφορετικές συστάδες στο C.

Διαχωρισμός του C^* = μήκος d^* της $(k-1)$-ης μακρύτερης ακμές στο ελάχιστο γεννητικό δέντρο. Η ακμή (p, q) έχει μήκος $\leq d^*$ αφού προστέθηκε από τον Kruskal.

Ο διαχωρισμός του C είναι $\leq d^*$ αφού τα p και q είναι σε διαφορετικές συστάδες. ▪

Αυτή είναι η ακμή που θα είχε προσθέσει ο Kruskal αν δεν τον είχαμε τερματίσει.



Οι ακμές που απομένουν μετά τη διαγραφή των $k-1$ μακρύτερων ακμών από το ελάχιστο γεννητικό δέντρο.