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/

Η τεχνική διαίρει και βασίλευε

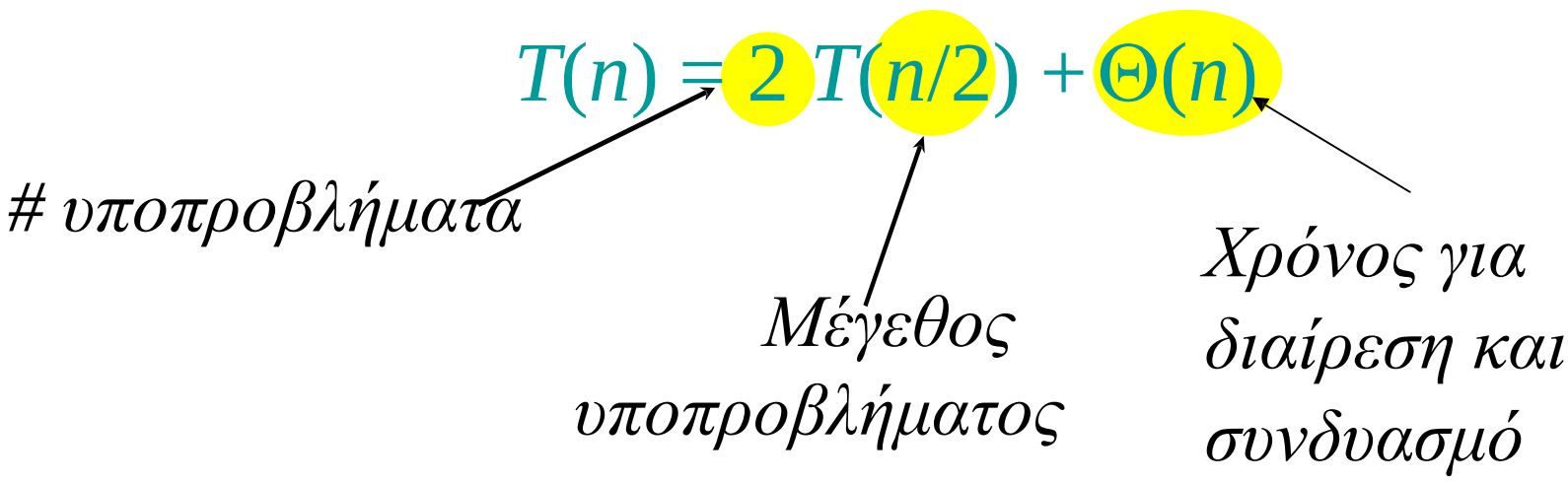
1. ***Διαίρεσε*** το πρόβλημα σε υποπροβλήματα
2. ***Βασίλευε*** τα υποπροβλήματα επιλύοντας τα αναδρομικά.
3. ***Συνδύασε*** τις λύσεις των υποπροβλημάτων.

Συγχωνευτική ταξινόμηση

1. ***Διαίρεσε:*** Τετριμμένο.
2. ***Βασίλευε:*** Αναδρομικά ταξινόμησε τους 2 υποπίνακες.
3. ***Συνδύασε:*** Συγχώνευση γραμμικού χρόνου.

Συγχωνευτική ταξινόμηση

1. ***Διαίρεσε:*** Τετριμμένο.
2. ***Βασίλευε:*** Αναδρομικά ταξινόμησε τους 2 υποπίνακες.
3. ***Συνδύασε:*** Συγχώνευση γραμμικού χρόνου



Θεώρημα κυρίαρχου όρου

$$T(n) = a\,T(n/b) + f(n)$$

Περίπτωση 1: $f(n) = O(n^{\log_b a - \varepsilon})$, σταθερά $\varepsilon > 0$

$\Rightarrow T(n) = \Theta(n^{\log_b a})$.

Περίπτωση 2: $f(n) = \Theta(n^{\log_b a} \lg^k n)$, σταθερά $k \geq 0$

$\Rightarrow T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$.

Περίπτωση 3: $f(n) = \Omega(n^{\log_b a + \varepsilon})$, σταθερά $\varepsilon > 0$, και συνθήκη κανονικότητας

$\Rightarrow T(n) = \Theta(f(n))$.

Συγχ. Ταξιν.: $a = 2,\ b = 2 \Rightarrow\ n^{\log_b a} = n^{\log_2 2} = n$

$\Rightarrow$ Περίπτωση 2 $(k = 0) \Rightarrow\ T(n) = \Theta(n \lg n)$.

Δυαδική αναζήτηση

Βρες ένα στοιχείο σε ένα ταξινομημένο πίνακα:

1. ***Διαίρεσε:*** Έλεγξε το μεσαίο στοιχείο
2. ***Βασίλευε:*** Αναδρομικά αναζήτησε σε 1 υποπίνακα.
3. ***Συνδύασε:*** Τετριμμένο.

Example: Εύρεση του 9

3	5	7	8	9	12	15

Δυαδική αναζήτηση

Βρες ένα στοιχείο σε ένα ταξινομημένο πίνακα:

1. ***Διαίρεσε:*** Έλεγξε το μεσαίο στοιχείο
2. ***Βασίλευε:*** Αναδρομικά αναζήτησε σε 1 υποπίνακα.
3. ***Συνδύασε:*** Τετριμμένο.

Example: Εύρεση του 9

Δυαδική Αναζήτηση

Βρες ένα στοιχείο σε ένα ταξινομημένο πίνακα:

1. ***Διαίρεσε:*** Έλεγξε το μεσαίο στοιχείο
2. ***Βασίλευε:*** Αναδρομικά αναζήτησε σε 1 υποπίνακα.
3. ***Συνδύασε:*** Τετριμμένο.

Example: Εύρεση του 9

3	5	7	8	9	12	15

Δυαδική Αναζήτηση

Βρες ένα στοιχείο σε ένα ταξινομημένο πίνακα:

1. ***Διαίρεσε:*** Έλεγξε το μεσαίο στοιχείο
2. ***Βασίλευε:*** Αναδρομικά αναζήτησε σε 1 υποπίνακα.
3. ***Συνδύασε:*** Τετριμμένο.

Example: Εύρεση του 9

3 5 7 8 9 12 15

Binary search

Βρες ένα στοιχείο σε ένα ταξινομημένο πίνακα:

1. ***Διαίρεσε:*** Έλεγξε το μεσαίο στοιχείο
2. ***Βασίλευε:*** Αναδρομικά αναζήτησε σε 1 υποπίνακα.
3. ***Συνδύασε:*** Τετριμμένο.

Example: Εύρεση του 9

3 5 7 8 9 12 15

Αναδρομή για δυαδική αναζήτηση

$$T(n) = 1\,T(n/2) + \Theta(1)$$

 # υποπροβλήματα

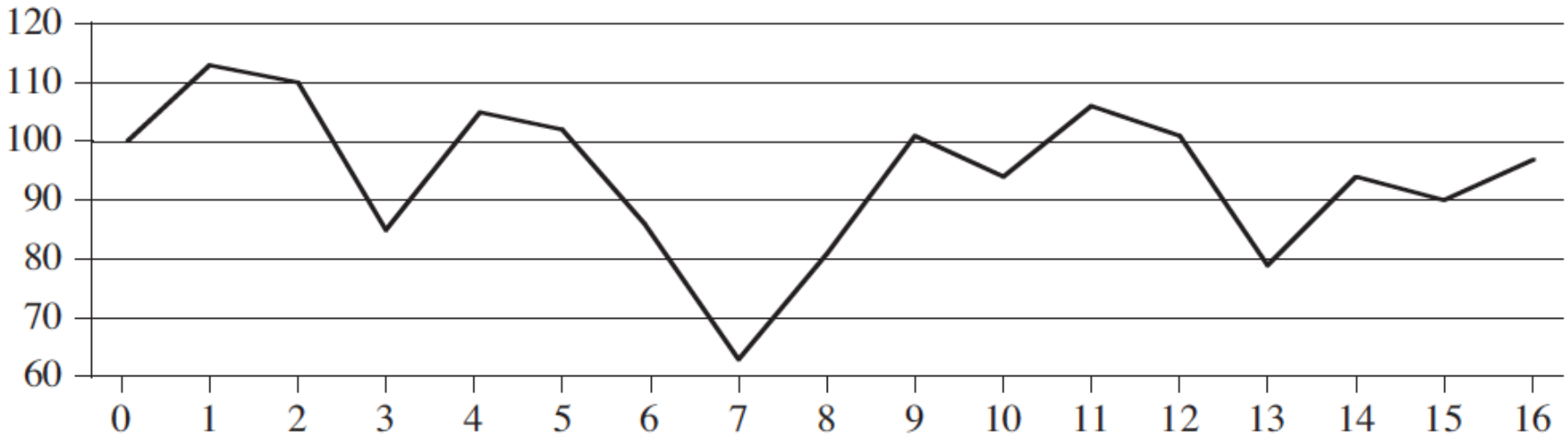
Μέγεθος υποπροβλήματος

Εργασία διαίρεσης και συνδυασμού

$n^{\log_b a} = n^{\log_2 1} = n^0 = 1 \Rightarrow$ Περίπτωση 2 ($k = 0$)

$\Rightarrow T(n) = \Theta(\lg n)$.

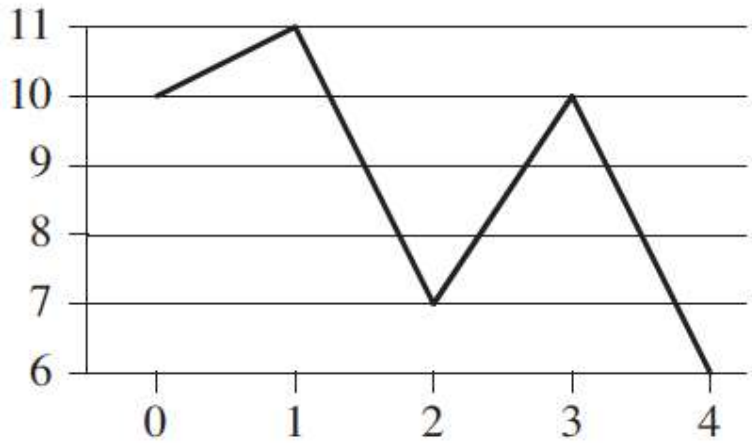
Το πρόβλημα της μέγιστης υποσυστοιχίας



Ημέρα	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Τιμή	100	113	110	85	105	102	86	63	81	101	94	106	101	79	94	90	97
Μεταβολή		13	−3	−25	20	−3	−16	−23	18	20	−7	12	−5	−22	15	−4	7

Σχήμα 4.1 Πληροφορίες για την τιμή των μετοχών της εταιρείας Πτητικά Χημικά Α.Ε. μετά το τέλος της κάθε συνεδρίασης για μια περίοδο 17 ημερών. Στον οριζόντιο άξονα του διαγράμματος απεικονίζεται η ημέρα, και στον κατακόρυφο η τιμή. Στην τελευταία γραμμή του πίνακα παρατίθεται η μεταβολή της τιμής κάθε μέρα σε σχέση με την προηγούμενη μέρα.

Το πρόβλημα της μέγιστης υποσυστοιχίας



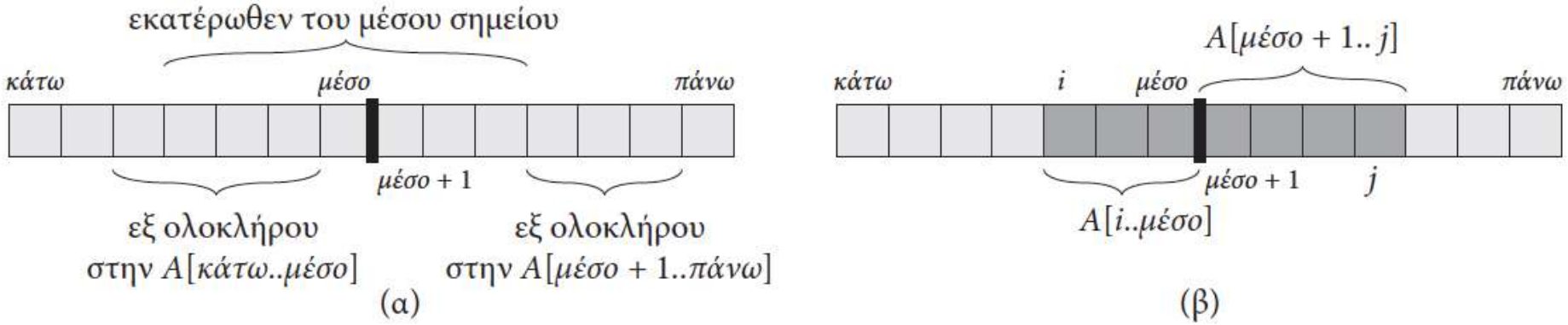
Ημέρα	0	1	2	3	4
Τιμή	10	11	7	10	6
Μεταβολή		1	−4	3	−4

Σχήμα 4.2 Ένα παράδειγμα που δείχνει ότι το μέγιστο κέρδος δεν εξασφαλίζεται πάντα αν ξεκινήσουμε από τη χαμηλότερη τιμή ή αν σταματήσουμε στην ψηλότερη. Και πάλι, στον οριζόντιο άξονα απεικονίζεται η ημέρα και στην κατακόρυφο η τιμή. Στην προκειμένη περίπτωση το μέγιστο κέρδος, 3 ευρώ ανά μετοχή, προκύπτει αν αγοράσουμε μετά την ημέρα 2 και πουλήσουμε μετά την ημέρα 3. Η τιμή 7 ευρώ μετά την ημέρα 2 δεν είναι η χαμηλότερη όλων, και η τιμή 10 ευρώ μετά την ημέρα 3 δεν είναι η ψηλότερη όλων.

Το πρόβλημα της μέγιστης υποσυστοιχίας

Σχήμα 4.3 Η μεταβολή της τιμής των μετοχών σαν ένα πρόβλημα μέγιστης υποσυστοιχίας. Στην προκειμένη περίπτωση, η υποσυστοιχία που έχει το μεγαλύτερο άθροισμα απ' όλες τις συνεχείς υποσυστοιχίες της A είναι η $A[8\,..\,11]$, με άθροισμα 43.

Το πρόβλημα της μέγιστης υποσυστοιχίας



Σχήμα 4.4 (α) Πιθανές θέσεις των υποσυστοιχιών της $A[κάτω\,..\,πάνω]$: εξ ολοκλήρου στην $A[κάτω\,..\,μέσο]$, εξ ολοκλήρου στην $A[μέσο + 1\,..\,πάνω]$, και εκατέρωθεν του μέσου σημείου *μέσο*. (β) Οποιαδήποτε υποσυστοιχία της $A[κάτω\,..\,πάνω]$ που βρίσκεται εκατέρωθεν του μέσου σημείου αποτελείται από δύο υποσυστοιχίες $A[i\,..\,μέσο]$ και $A[μέσο + 1\,..\,j]$, όπου $κάτω \leq i \leq μέσο$ και $μέσο < j \leq πάνω$.

Το πρόβλημα της μέγιστης υποσυστοιχίας

```
FIND-MAX-CROSSING-SUBARRAY(A, low, mid, high)
 1  left-sum = −∞
 2  sum = 0
 3  for i = mid downto low
 4       sum = sum + A[i]
 5       if sum > left-sum
 6            left-sum = sum
 7            max-left = i
 8  right-sum = −∞
 9  sum = 0
10  for j = mid + 1 to high
11       sum = sum + A[j]
12       if sum > right-sum
13            right-sum = sum
14            max-right = j
15  return (max-left, max-right, left-sum + right-sum)
```

Το πρόβλημα της μέγιστης υποσυστοιχίας

```
FIND-MAXIMUM-SUBARRAY(A, low, high)
 1  if high == low
 2       return (low, high, A[low])          // base case: only one element
 3  else mid = ⌊(low + high)/2⌋
 4       (left-low, left-high, left-sum) =
              FIND-MAXIMUM-SUBARRAY(A, low, mid)
 5       (right-low, right-high, right-sum) =
              FIND-MAXIMUM-SUBARRAY(A, mid + 1, high)
 6       (cross-low, cross-high, cross-sum) =
              FIND-MAX-CROSSING-SUBARRAY(A, low, mid, high)
 7       if left-sum ≥ right-sum and left-sum ≥ cross-sum
 8            return (left-low, left-high, left-sum)
 9       elseif right-sum ≥ left-sum and right-sum ≥ cross-sum
10            return (right-low, right-high, right-sum)
11       else return (cross-low, cross-high, cross-sum)
```

Το πρόβλημα της μέγιστης υποσυστοιχίας

Ο χρόνος εκτέλεσης του «Διαίρει και Βασίλευε» Αλγορίθμου:

$$T(n) = \begin{cases} \Theta(1) \ \ \alpha\nu\ n = 1 \\ 2T\left(\frac{n}{2}\right) + \Theta(n)\ \alpha\nu\ n > 1 \end{cases}$$

$$T\,(n) = \Theta(n\ lg\ n)$$

Μέτρηση Αναστροφών

Ένας ιστότοπος μουσικής προσπαθεί να ταιριάξει τις προτιμήσεις σου σε τραγούδια με αυτές άλλων.

- Εσύ κατατάσσεις n τραγούδια.
- Ο ιστότοπος συμβουλεύεται τη βάση δεδομένων για την εύρεση ατόμων με παρόμοιες προτιμήσεις.

Μέτρο ομοιότητας: πλήθος αναστροφών μεταξύ δύο κατατάξεων.

- Η κατάταξή μου: $1, 2, \ldots, n$.
- Η κατάταξή σου: $a_1, a_2, \ldots, a_n$.
- Τα ταξίδια i και j είναι ανεστραμμένα αν $i < j$, αλλά $a_i > a_j$.

	A	B	C	D	E
me	1	2	3	4	5
you	1	3	4	2	5

2 inversions: 3-2, 4-2

Ωμή βία: έλεγξε όλα τα ζεύγη, $\Theta(n^2)$ σε πλήθος.

Μέτρηση Αναστροφών: διαίρει-και-βασίλευε

- Διαίρεσε: χώρισε τη λίστα σε δύο μισά A και B.
- Βασίλευε:αναδρομικά μέτρησε τις αναστροφές σε κάθε λίστα.
- Συνδύασε: μέτρησε τις αναστροφές (a, b) με $a \in A$ και $b \in B$.
  Επέστρεψε το άθροισμα των τριών μετρητών.

input

1	5	4	8	10	2	6	9	3	7

count inversions in left half A

1	5	4	8	10

5–4

count inversions in right half B

2	6	9	3	7

6–3 9–3 9–7

count inversions (a, b) with a $\in$ A and b $\in$ B

1	5	4	8	10

2	6	9	3	7

4–2 4–3 5–2 5–3 8–2 8–3 8–6 8–7 10–2 10–3 10–6 10–7 10–9

output 1 + 3 + 13 = 17

Counting inversions: how to combine two subproblems?

Q. How to count inversions (a, b) with $a \in A$ and $b \in B$?

A. Easy if A and B are sorted!

Warmup algorithm.

- Sort A and B.
- For each element $b \in B$,
  - binary search in A to find how elements in A are greater than b.

list A

7	10	18	3	14

list B

20	23	2	11	16

sort A

3	7	10	14	18

sort B

2	11	16	20	23

binary search to count inversions (a, b) with a ∈ A and b ∈ B

3	7	10	14	18

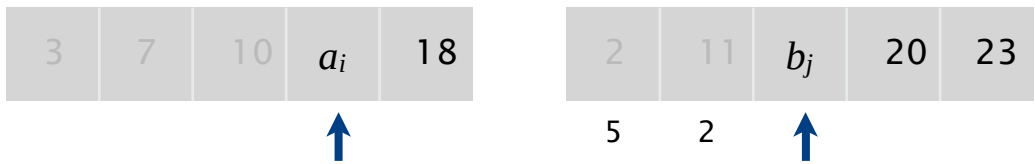
2	11	16	20	23
5	2	1	0	0

Μέτρηση Αναστροφών: πως να συνδυαστούν δύο υποπροβλήματα;

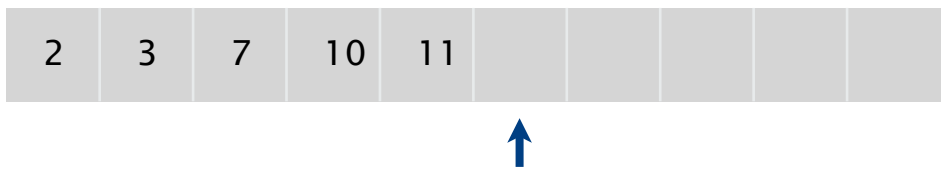
Μέτρησε τις αναστροφές (a, b) με $a \in A$ και $b \in B$, υποθέτοντας ότι τα A και B είναι ταξινομημένα.

- Διάβασε τα A και B από αριστερά προς τα δεξιά.
- Σύγκρινε το a_i και b_j.
- Αν $a_i < b_j$, τότε a_i δεν είναι ανεστραμμένο με κάποιο από τα στοιχεία που απέμειναν στο B.
- Αν $a_i > b_j$, τότε b_j είναι ανεστραμμένο με κάθε ένα από τα στοιχεία που έχουν απομείνει στο A.
- Τοποθέτησε το μικρότερο στοιχείο στην ταξινομημένη λίστα C.

count inversions (a, b) with a ∈ A and b ∈ B



merge to form sorted list C



Μέτρηση Αναστροφών: υλοποίηση του διαίρει και βασίλευε αλγορίθμου

Είσοδος. Λίστα L.

Έξοδος. Πλήθος των αναστροφών στην L και L ταξινομημένη.

SORT-AND-COUNT(L)

IF (list L has one element)

  RETURN $(0, L)$.

Divide the list into two halves A and B.

$(r_A, A) \leftarrow$ SORT-AND-COUNT(A). ⟵ $T(n / 2)$

$(r_B, B) \leftarrow$ SORT-AND-COUNT(B). ⟵ $T(n / 2)$

$(r_{AB}, L) \leftarrow$ MERGE-AND-COUNT(A, B). ⟵ $\Theta(n)$

RETURN $(r_A + r_B + r_{AB}, L)$.

Μέτρηση Αναστροφών: ανάλυση του διαίρει και βασίλευε αλγορίθμου

Πρόταση. Ο αλγόριθμος sort-and-count μετράει το πλήθος των αναστροφών σε μία μετάθεση μεγέθους n σε χρόνο $O(n \log n)$.

Απ. Ο χρόνος χειρότερης περίπτωσης $T(n)$ ικανοποιεί την αναδρομή:

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + \Theta(n) & \text{if } n > 1 \end{cases}$$

Ύψωση σε δύναμη

Πρόβλημα: Υπολόγισε a^n, όπου $n \in N$.

Απλός Αλγόριθμος: $\Theta(n)$.

Ύψωση σε δύναμη

Πρόβλημα: Υπολόγισε a^n, όπου $n \in N$.

Απλός Αλγόριθμος: $\Theta(n)$.

Αλγόριθμος Διαίρει και Βασίλευε:

$$a^n = \begin{cases} a^{n/2} \cdot a^{n/2} & \text{αν } n \text{ is άρτιος} \\ a^{(n-1)/2} \cdot a^{(n-1)/2} \cdot a & \text{αν } n \text{ είναι περιττός.} \end{cases}$$

$$T(n) = T(n/2) + \Theta(1) \Rightarrow T(n) = \Theta(\lg n).$$

Πολλαπλασιασμός Πινάκων

Είσοδος: $A = [a_{ij}], B = [b_{ij}].$
Έξοδος: $C = [c_{ij}] = A \cdot B.$
$\Big\}$ $i, j = 1, 2, \ldots, n.$

$$\begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{bmatrix}$$

$$c_{ij} = \sum_{k=1}^{n} a_{ik} \cdot b_{kj}$$

Ο συνήθης αλγόριθμος

for $i \leftarrow 1$ **to** n
  do for $j \leftarrow 1$ **to** n
    do $c_{ij} \leftarrow 0$
      for $k \leftarrow 1$ **to** n
        do $c_{ij} \leftarrow c_{ij} + a_{ik} \cdot b_{kj}$

Χρόνος εκτέλεσης = $\Theta(n^3)$

Διαίρει και Βασίλευε Αλγόριθμος

Ιδέα:

$n \times n$ πίνακας = 2×2 πίνακας $(n/2) \times (n/2)$ υποπινάκων:

$$\begin{bmatrix} r & s \\ t & u \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} e & f \\ g & h \end{bmatrix}$$

$$C = A \cdot B$$

$r = ae + bg$
$s = af + bh$
$t = ce + dg$
$u = cf + dh$

αναδρομή

8 πολ/σμοι $(n/2) \times (n/2)$ υποπινάκων

4 προσθ. $(n/2) \times (n/2)$ υποπινάκων

Ανάλυση του ΔκΒ αλγορίθμου

$$T(n) = 8\,T(n/2) + \Theta(n^2)$$

 # υποπίνακες

Μέγεθος υποπίνακα

Χρόνος για την πρόσθεση των υποπινάκων

$n^{\log_b a} = n^{\log_2 8} = n^3 \Rightarrow$ Περίπτωση 1 $\Rightarrow T(n) = \Theta(n^3)$.

Δεν είναι καλύτερος από το συνηθισμένο αλγόριθμο.

Η Ιδέα του Strassen

- Πολλαπλασίασε 2×2 πίνακες με μόνο 7 αναδρομικούς πολ/σμους.

$$P_1 = a \cdot (f - h)$$
$$P_2 = (a + b) \cdot h$$
$$P_3 = (c + d) \cdot e$$
$$P_4 = d \cdot (g - e)$$
$$P_5 = (a + d) \cdot (e + h)$$
$$P_6 = (b - d) \cdot (g + h)$$
$$P_7 = (a - c) \cdot (e + f)$$

$$r = P_5 + P_4 - P_2 + P_6$$
$$s = P_1 + P_2$$
$$t = P_3 + P_4$$
$$u = P_5 + P_1 - P_3 - P_7$$

7 πολ/σμους,
18 πρσθ./αφαιρ.

Η Ιδέα του Strassen

- Πολλαπλασίασε 2×2 matrices με μόνο 7 αναδρομικούς πολ/σμους.

$$P_1 = a \cdot (f - h)$$
$$P_2 = (a + b) \cdot h$$
$$P_3 = (c + d) \cdot e$$
$$P_4 = d \cdot (g - e)$$
$$P_5 = (a + d) \cdot (e + h)$$
$$P_6 = (b - d) \cdot (g + h)$$
$$P_7 = (a - c) \cdot (e + f)$$

$$\begin{aligned} r &= P_5 + P_4 - P_2 + P_6 \\ &= (a + d)(e + h) \\ &\quad + d(g - e) - (a + b)h \\ &\quad + (b - d)(g + h) \\ &= ae + ah + de + dh \\ &\quad + dg - de - ah - bh \\ &\quad + bg + bh - dg - dh \\ &= ae + bg \end{aligned}$$

Ο αλγόριθμος του Strassen

1. ***Διαίρεσε:*** Διαμέρισε A και B σε $(n/2)\times(n/2)$ υποπίνακες. Σχημάτισε τους όρους που θα πολλαπλασιαστούν χρησιμοποιώντας $+$ και $-$.
2. ***Βασίλευε:*** Εκτέλεσε 7 πολλαπλασιασμούς $(n/2)\times(n/2)$ υποπινάκων αναδρομικά.
3. ***Συνδύασε:*** Υπολόγισε το C χρησιμοποιώντας $+$ και $-$ στους $(n/2)\times(n/2)$ υποπίνακες.

$$T(n) = 7\,T(n/2) + \Theta(n^2)$$

Ανάλυση του Αλγορίθμου του Strassen

$$T(n) = 7\,T(n/2) + \Theta(n^2)$$

$n^{\log_b a} = n^{\log_2 7} \approx n^{2.81} \Rightarrow$ Περίπτωση 1 $\Rightarrow T(n) = \Theta(n^{\lg 7})$.

Ο αριθμός 2.81 μπορεί να μην φαίνεται πολύ μικρότερος του 3, αλλά επειδή η διαφορά είναι στον εκθέτη, η επίδραση στο χρόνο εκτέλεσης είναι σημαντική.

Στην πράξη, ο αλγόριθμος του Strassen είναι καλύτερος από τον απλό αλγόριθμο σε σημερινές αρχιτεκτονικές για $n \geq 32$ περίπου.

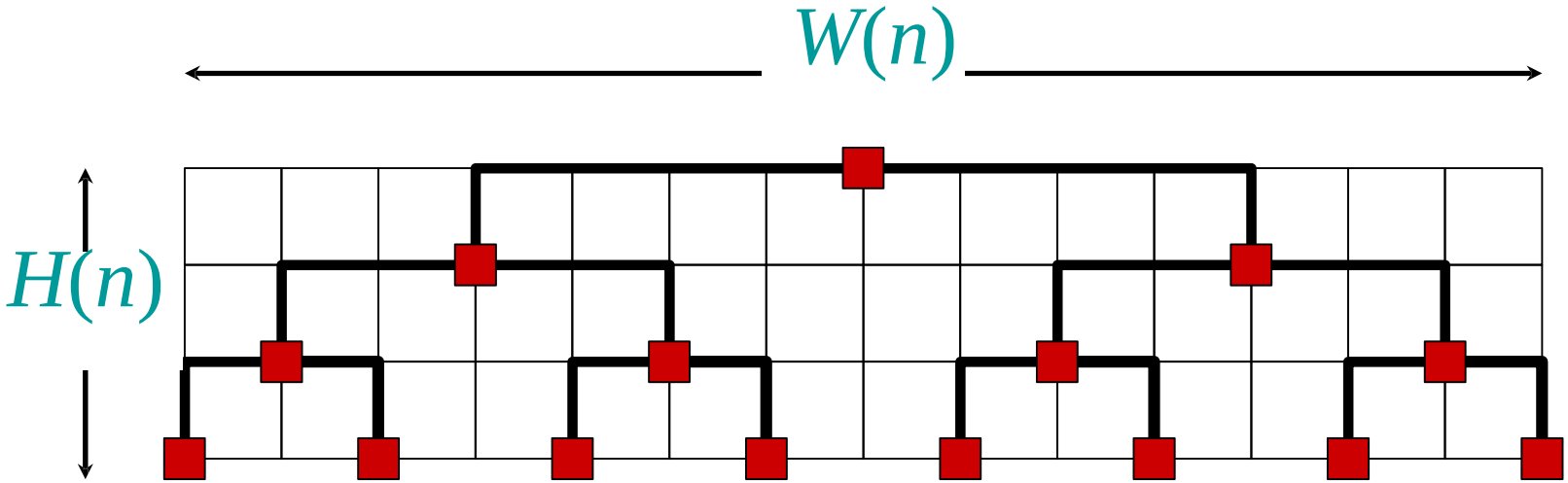
Υπάρχει και καλύτερος αλγόριθμος (μόνο θεωρητικού ενδιαφέροντος): $\Theta(n^{2.376\ldots})$.

Τοποθέτηση VLSI

Πρόβλημα: Τοποθέτησε ένα πλήρες δυαδικό δένδρο με n φύλλα σε ένα πλέγμα με ελάχιστο εμβαδό.

Τοποθέτηση VLSI

Πρόβλημα: Τοποθέτησε ένα πλήρες δυαδικό δένδρο με n φύλλα σε ένα πλέγμα με ελάχιστο εμβαδό.

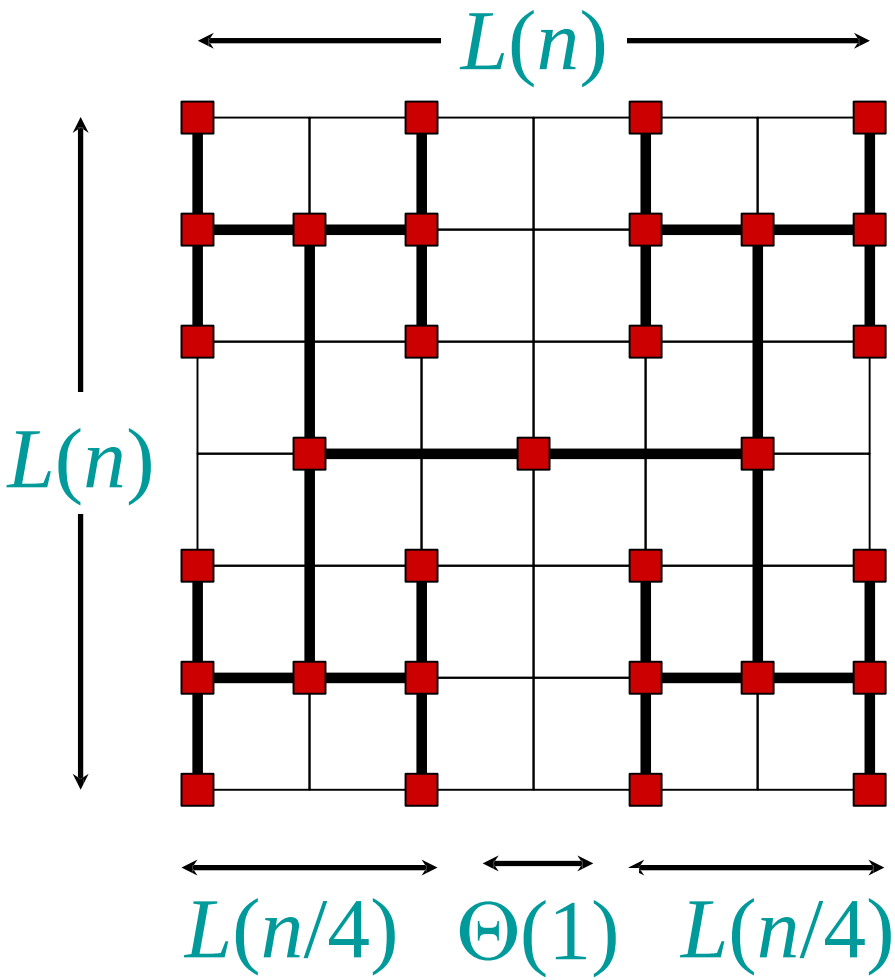


$H(n) = H(n/2) + \Theta(1)$
$= \Theta(\lg n)$

$W(n) = 2\,W(n/2) + \Theta(1)$
$= \Theta(n)$

Εμβαδό $= \Theta(n \lg n)$

Τοποθέτηση H-tree



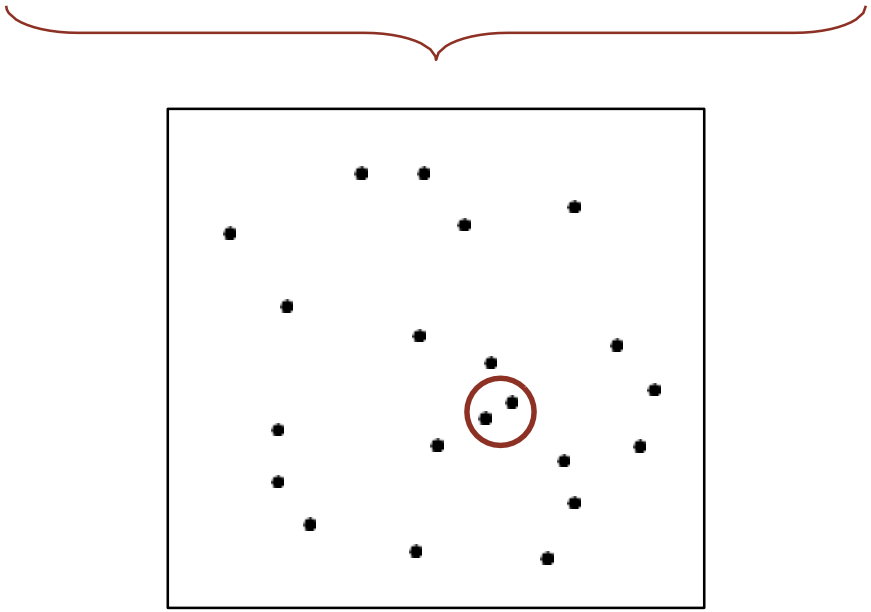
$$L(n) = 2\,L(n/4) + \Theta(1)$$
$$= \Theta(\sqrt{n})$$

Εμβαδό $= \Theta(n)$

Πλησιέστερο ζεύγος σημείων

Πρόβλημα πλησιέστερου ζεύγους σημείων. Δοθέντων n σημείων στο επίπεδο, βρες ένα ζεύγος σημείων με τη μικρότερη ευκλείδεια απόσταση μεταξύ αυτών.

Βασική λειτουργία στο χώρο της γεωμετρίας.



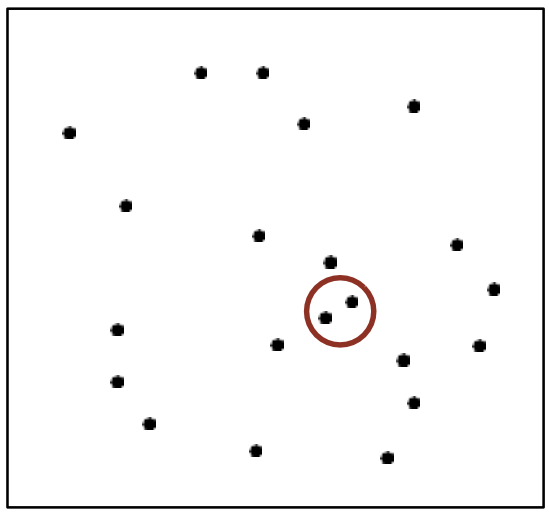
Πλησιέστερο ζεύγος σημείων

Πρόβλημα πλησιέστερου ζεύγους σημείων. Δοθέντων n σημείων στο επίπεδο, βρες ένα ζεύγος σημείων με τη μικρότερη ευκλείδεια απόσταση μεταξύ αυτών.

Ωμή βία. Έλεγξε όλα τα ζεύγη με $\Theta(n^2)$ υπολογισμούς απόστασης.

1Δ εκδοχή. Εύκολος $O(n \log n)$ αλγόριθμος αν τα σημεία είναι σε ευθεία.

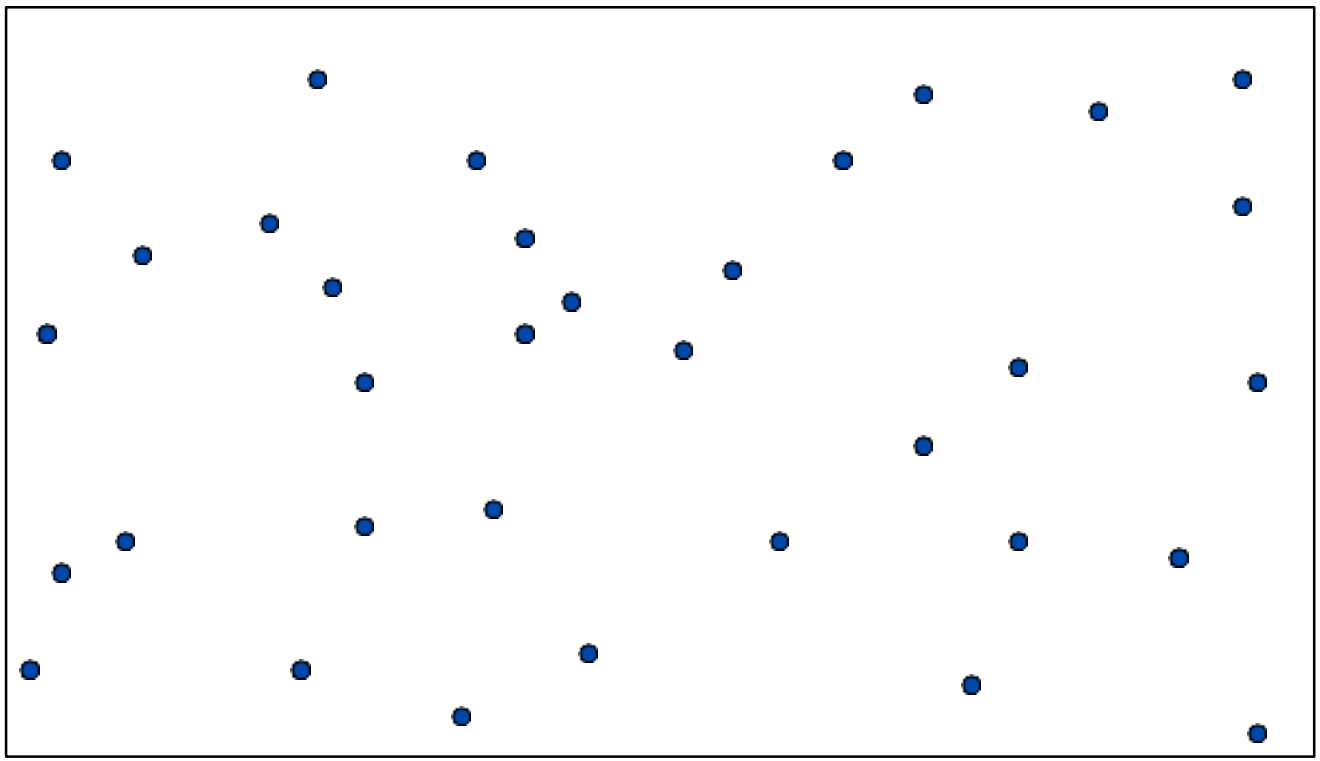
Υπόθεση. Δεν υπάρχουν δύο σημεία με την ίδια x-συντεταγμένη.



Πλησιέστερο ζεύγος σημείων: πρώτη προσπάθεια

Λύση βάσει ταξινόμησης.

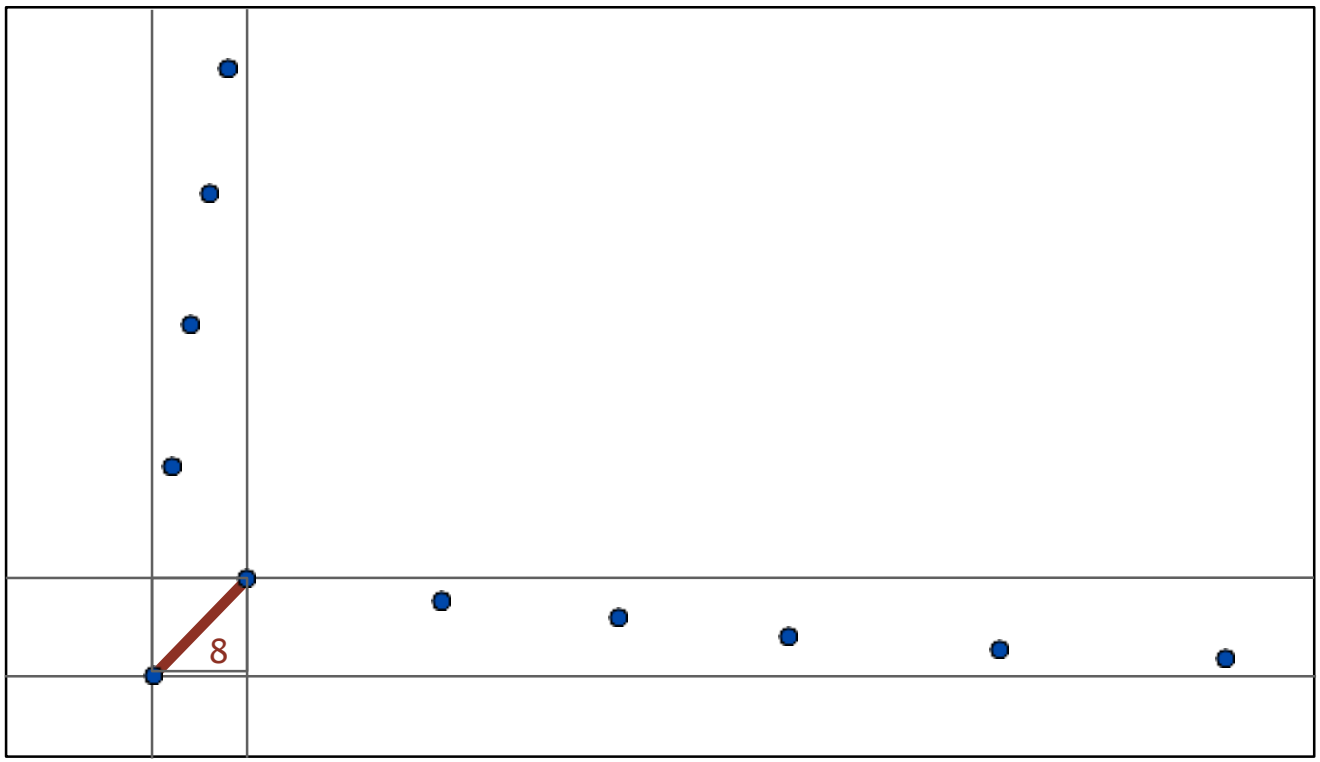
- Ταξινόμησε ως προς τη x-συντεταγμένη and θεώρησε κοντινά σημεία.
- Ταξινόμησε ως προς την y-συντεταγμένη και θεώρησε κοντινά σημεία.



Πλησιέστερο ζεύγος σημείων: πρώτη προσπάθεια

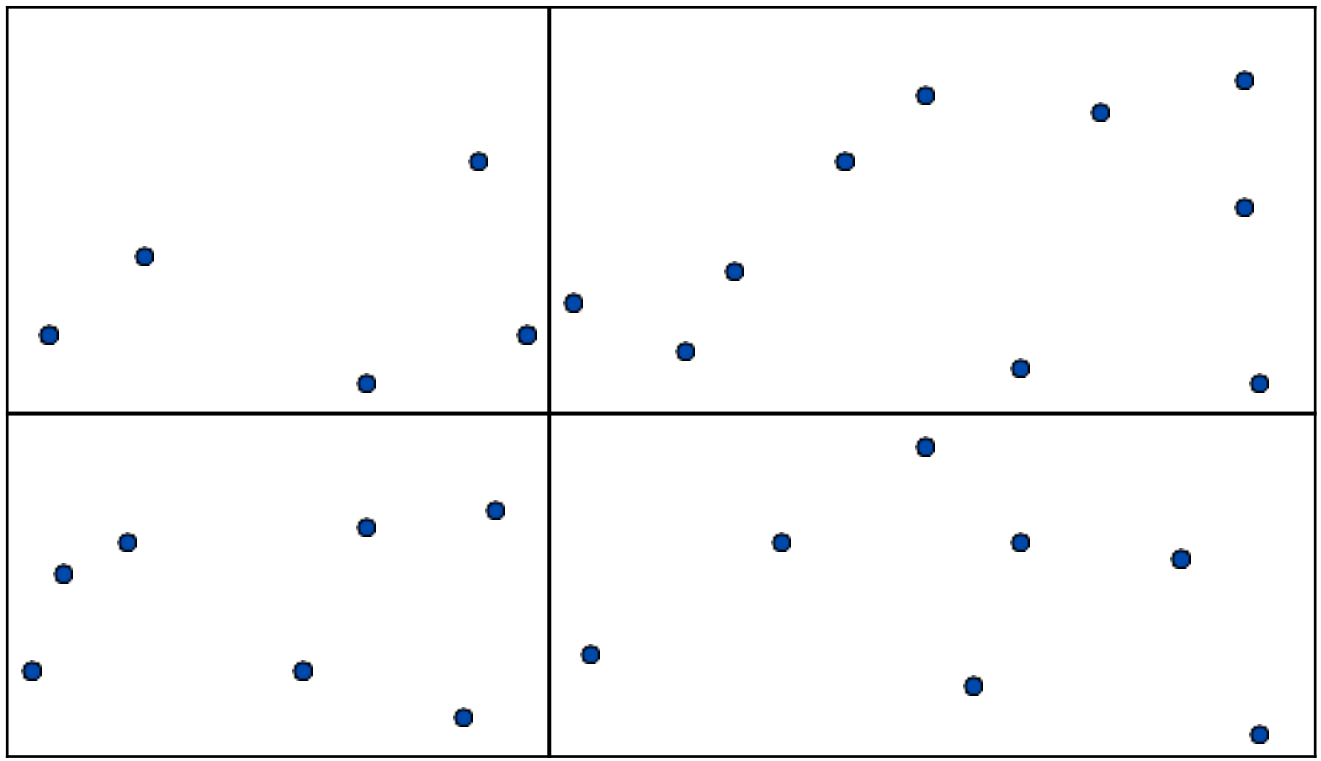
Λύση βάσει ταξινόμησης.

- Ταξινόμησε ως προς τη x-συντεταγμένη and θεώρησε κοντινά σημεία.
- Ταξινόμησε ως προς την y-συντεταγμένη και θεώρησε κοντινά σημεία.



Πλησιέστερο ζεύγος σημείων: δεύτερη προσπάθεια

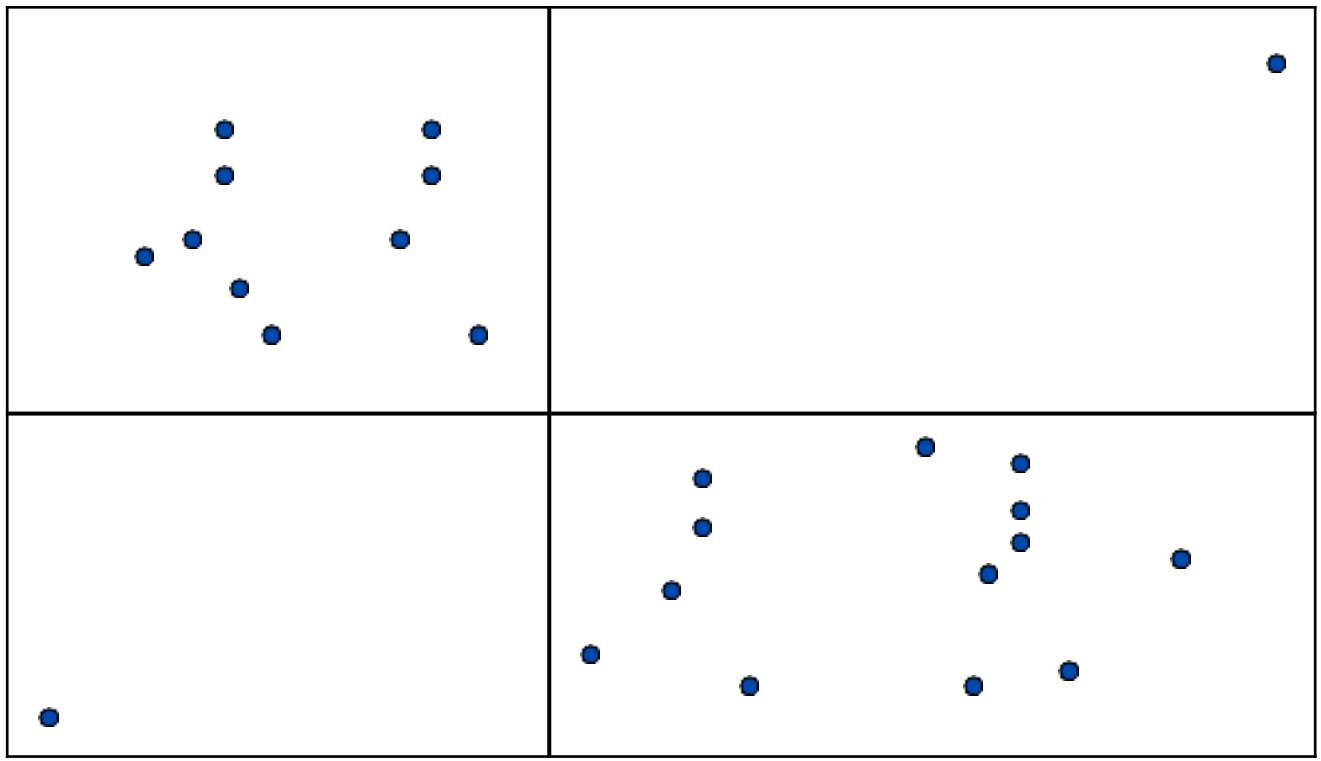
Διαίρεσε. Υποδιαίρεσε την περιοχή σε 4 τεταρτημόρια.



Πλησιέστερο ζεύγος σημείων: δεύτερη προσπάθεια

Διαίρεσε. Υποδιαίρεσε την περιοχή σε 4 τεταρτημόρια.

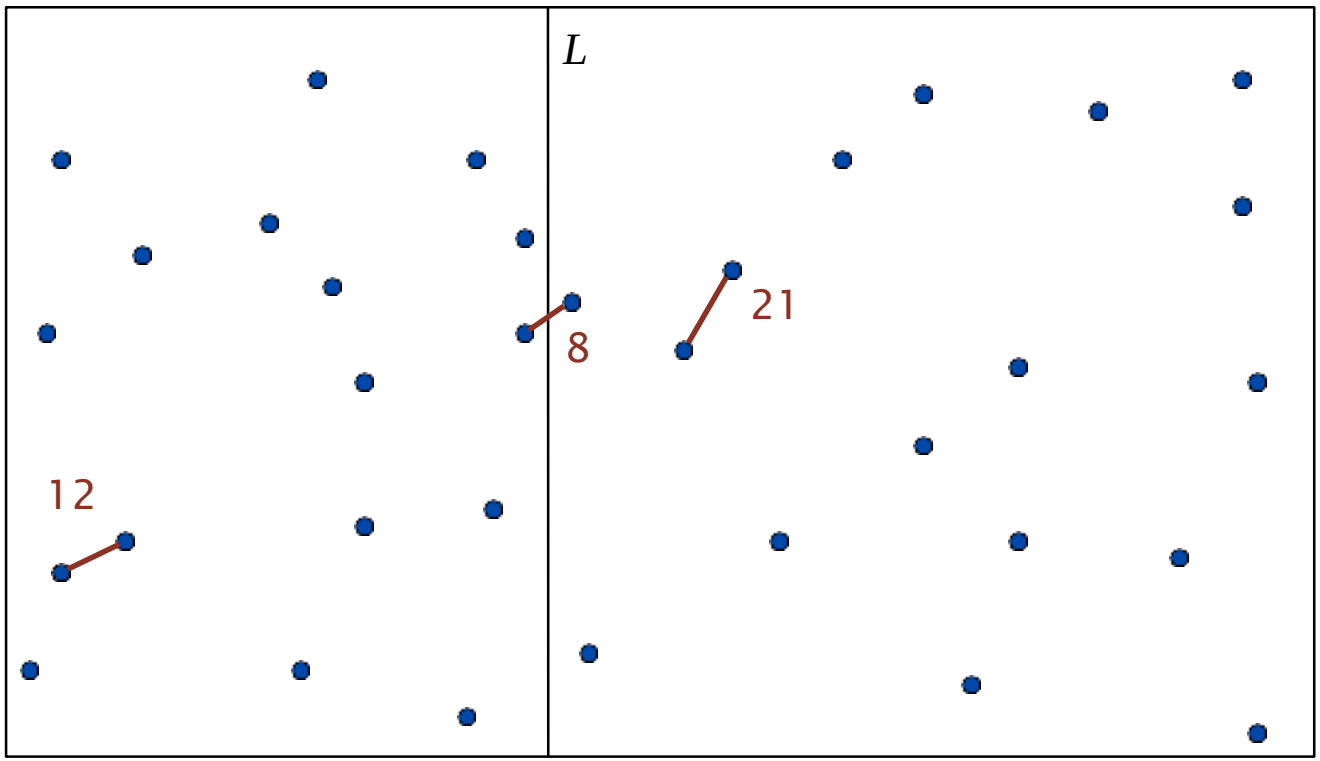
Εμπόδιο. Αδύνατο να εξασφαλιστούν $n/4$ σε κάθε τεταρτημόριο.



Πλησιέστερο ζεύγος σημείων: αλγόριθμος διαίρει και βασίλευε

- Διαίρεσε: σχεδίασε μία κάθετη γραμμή L έτσι ώστε $n/2$ σημεία σε κάθε πλευρά.
- Βασίλευε: Βρες το πλησιέστερο ζεύγος σε κάθε πλευρά αναδρομικά.
- Συνδύασε: βρες το πλησιέστερο ζεύγος με ένα σημείο σε κάθε πλευρά.
- Επίστρεψε την καλύτερη από τις 3 λύσεις.

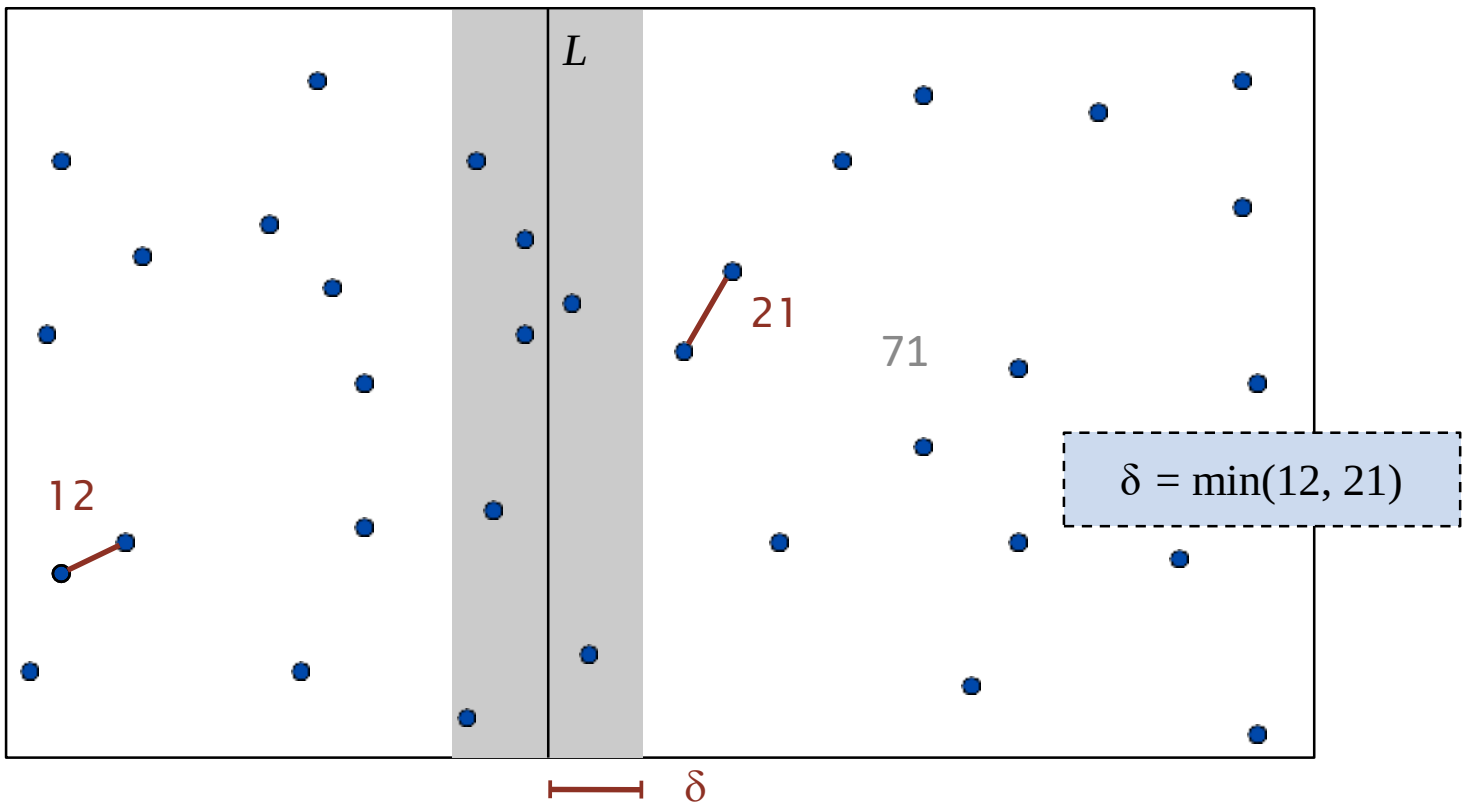
Φαίνεται ότι απαιτεί $\Theta(n^2)$



Πως θα βρω το πλησιέστερο ζεύγος με ένα σημείο σε κάθε πλευρά;

Βρες το πλησιέστερο ζεύγος με ένα σημείο σε κάθε πλευρά, υποθέτοντας ότι η απόσταση $< \delta$.

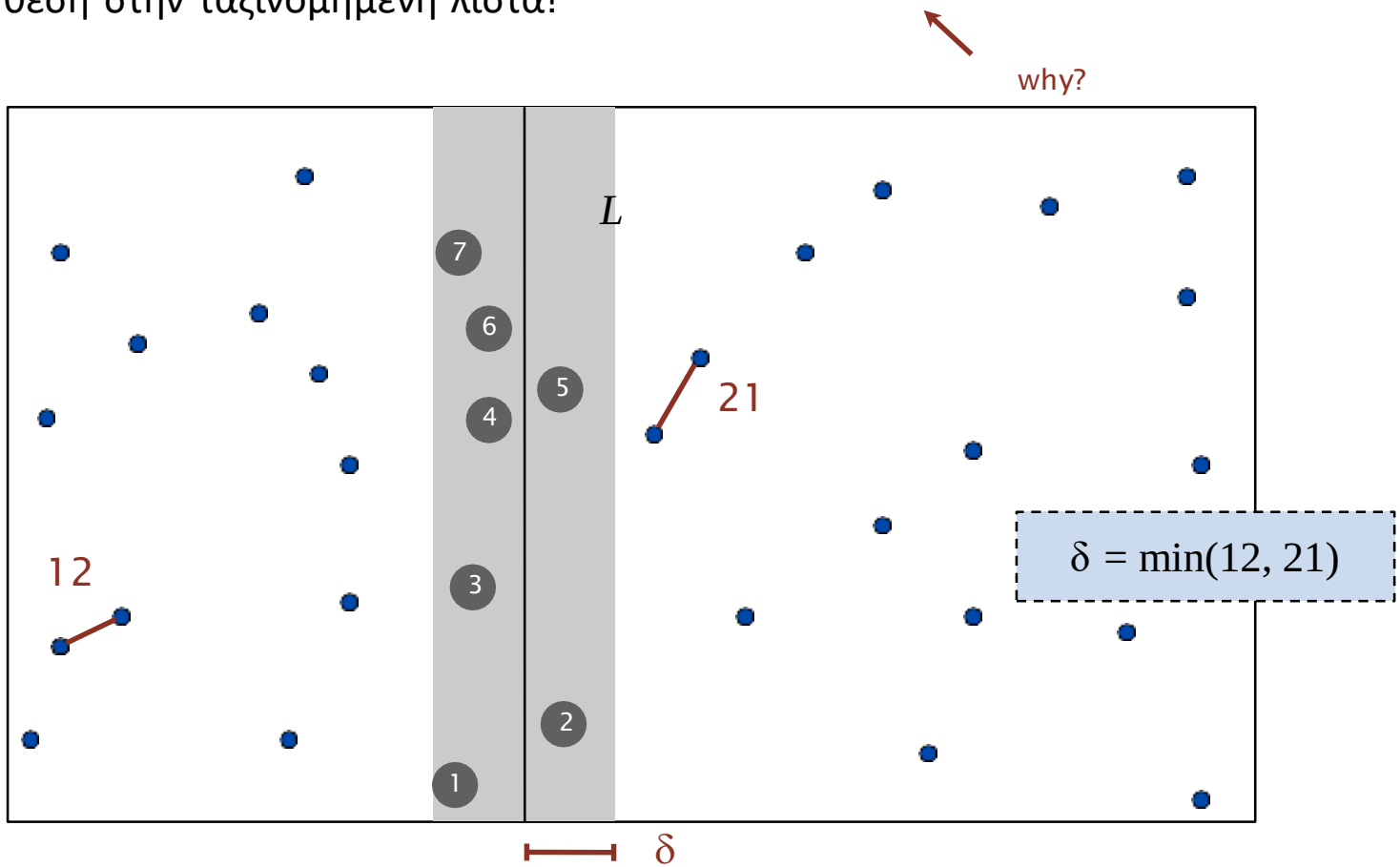
- Παρατήρηση: αρκεί να θεωρήσουμε μόνο αυτά εκείνα τα σημεία εντός απόστασης δ από τη γραμμή L.



Πως θα βρω το πλησιέστερο ζεύγος με ένα σημείο σε κάθε πλευρά;

Βρες το πλησιέστερο ζεύγος με ένα σημείο σε κάθε πλευρά, υποθέτοντας ότι η απόσταση $< \delta$.

- Παρατήρηση: αρκεί να θεωρήσουμε μόνο αυτά εκείνα τα σημεία εντός απόστασης δ από τη γραμμή L.
- Ταξινόμησε τα σημεία σε μία λωρίδα πλάτους 2δ ως προς τη y-συντεταγμένη.
- Έλεγξε τις αποστάσεις μόνο αυτών των σημείων που είναι εντός 7 θέσεων από την τρέχουσα θέση στην ταξινομημένη λίστα!



Πως θα βρω το πλησιέστερο ζεύγος με ένα σημείο σε κάθε πλευρά;

Ορ. Έστω s_i το σημείο στη λωρίδα πλάτους 2δ, με την $i^{οστη}$ μικρότερη y-συντεταγμένη.

Ισχυρισμός. Αν $|j - i| > 7$, τότε η απόσταση μεταξύ s_i και s_j είναι τουλάχιστον δ.

Απ.

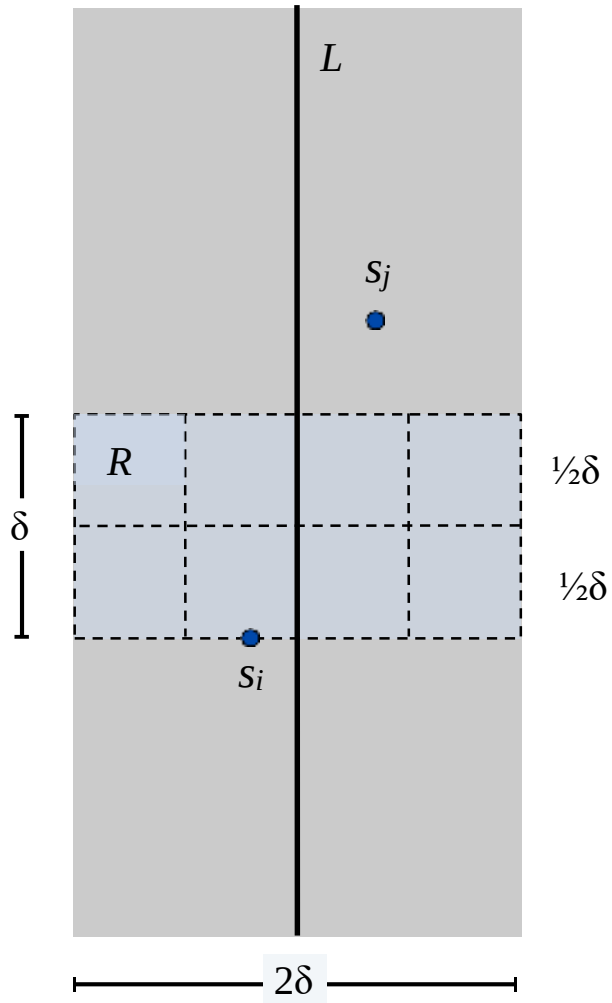
- Θεώρησε το 2δ X δ ορθογώνιο R σε μία λωρίδα της οποίας η ελάχιστη y-coordinate είναι η y-συντεταγμένη του s_i.
- Απόσταση μεταξύ s_i και κάθε σημείου s_j
  - Πάνω από το R is $\geq \delta$.

  Υποδιαίρεσε R σε 8 τετράγωνα.

diameter is $\delta / \sqrt{2} < \delta$

Το πολύ 1 σημείο ανά τετράγωνο.

Το πολύ 7 άλλα σημεία μπορεί να είναι στο R.



Closest pair of points: divide-and-conquer algorithm

CLOSEST-PAIR(p_1, p_2, …, p_n)

Compute vertical line L such that half the points are on each side of the line. ← $O(n)$

$\delta_1 \leftarrow$ CLOSEST-PAIR(points in left half). ← $T(n / 2)$

$\delta_2 \leftarrow$ CLOSEST-PAIR(points in right half). ← $T(n / 2)$

$\delta \leftarrow \min \{ \delta_1 , \delta_2 \}$.

Delete all points further than δ from line L. ← $O(n)$

Sort remaining points by y-coordinate. ← $O(n \log n)$

Scan points in y-order and compare distance between each point and next 7 neighbors. If any of these distances is less than δ, update δ. ← $O(n)$

RETURN δ.

Ποια είναι η λύση της ακόλουθης αναδρομής?

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + \Theta(n \log n) & \text{if } n > 1 \end{cases}$$

A. $T(n) = \Theta(n)$.

B. $T(n) = \Theta(n \log n)$.

C. $T(n) = \Theta(n \log^2 n)$.

D. $T(n) = \Theta(n^2)$.

Συμπεράσματα

- Η Διαίρει και Βασίλευε είναι μία από τις ισχυρές τεχνικές για σχεδίαση αλγορίθμων.
- Οι αλγόριθμοι Διαίρει και Βασίλευε μπορούν να αναλυθούν με τη χρήση αναδρομικών σχέσεων και το θεώρημα του κυρίαρχου όρου.
- Η στρατηγική συχνά οδηγεί σε αποδοτικούς αλγορίθμους.