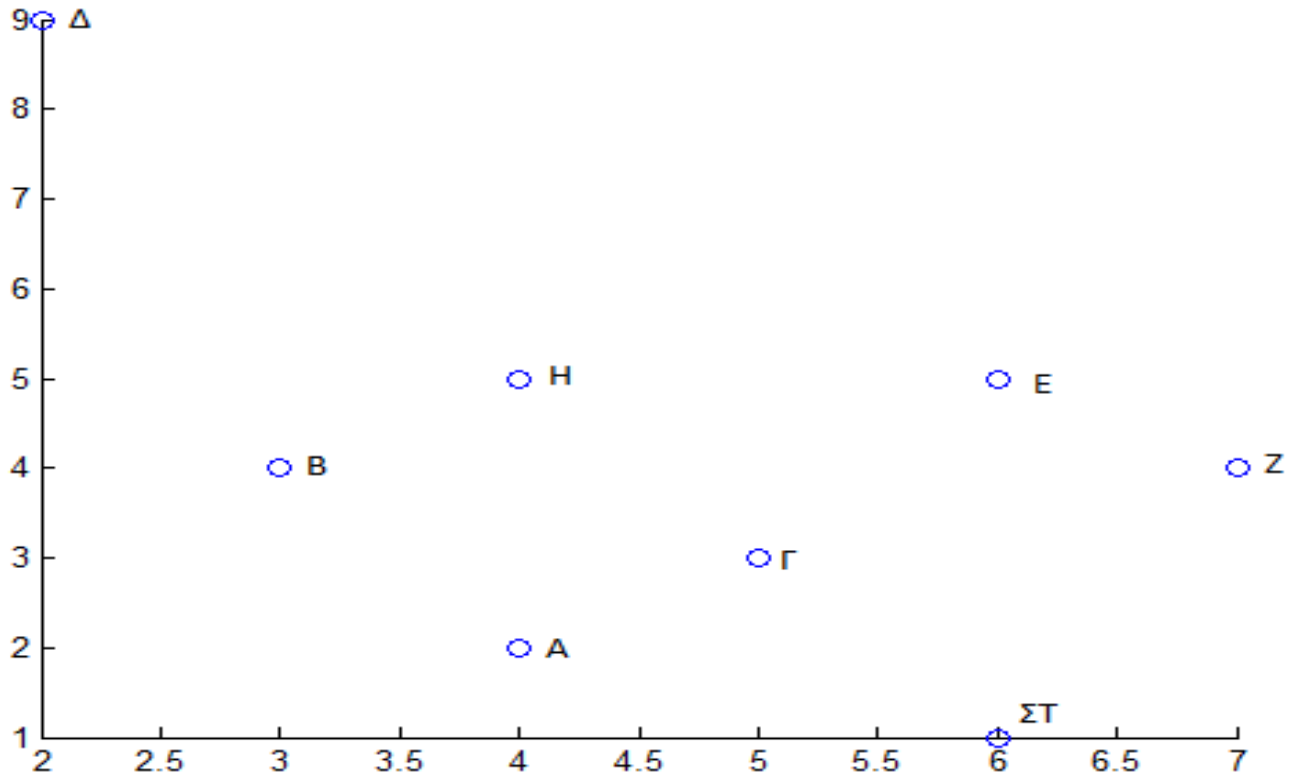
Άσκηση: Εύρεση Μέγιστων Σημείων

- Έστω ένα σύνολο $S = |n|$ σημείων του επιπέδου $\mathbb{R}^2$. Περιγράψτε έναν αποδοτικό αλγόριθμο για την εύρεση των μέγιστων σημείων του συνόλου S.
- Σημείωση: Ένα σημείο $p = (p_x, p_y)$ καλείται μέγιστο (maximal), εάν δεν κυριαρχείται (dominated) από κανένα άλλο σημείο. Ισοδύναμα, δεν υπάρχει άλλο σημείο $q = (q_x, q_y) \in S$, τέτοιο ώστε $p_x \leq q_x$ και $p_y \leq q_y$.

Εύρεση Μέγιστων Σημείων

Σημεία	X	Y
Α	4	2
Β	3	4
Γ	5	3
Δ	2	9
Ε	6	5
ΣΤ	6	1
Ζ	7	4
Η	4	5

Εύρεση Μέγιστων Σημείων



Εύρεση Μέγιστων Σημείων - Ταξινόμηση στοιχείων ως προς Χ

Λύση	Χ	Υ
Δ	2	9
Β	3	4
Α	4	2
Η	4	5
Γ	5	3
Ε	6	5
ΣΤ	6	1
Ζ	7	4

Εύρεση Μέγιστων Σημείων - Ταξινόμηση στοιχείων ως προς Χ

Λύση	Χ	Υ
Δ	2	9
Β	3	4
Α	4	2
Η	4	5
Γ	5	3
Ε	6	5
ΣΤ	6	1
Ζ	7	4

Βασική παρατήρηση:
για τα μέγιστα σημεία ισχύει ότι εφόσον είναι ταξινομημένα κατά αύξουσα σειρά τετμημένης, θα είναι εξ ορισμού ταξινομημένα κατά φθίνουσα σειρά τεταγμένης.

Εύρεση Μέγιστων Σημείων

Λύση	X	Y
Δ	2	9
Β	3	4
Α	4	2
Η	4	5
Γ	5	3
Ε	6	5
ΣΤ	6	1
Ζ	7	4

Εύρεση Μέγιστων Σημείων

Λύση	X	Y
Δ	2	9
Β	3	4
Α	4	2
Η	4	5
Γ	5	3
Ε	6	5
ΣΤ	6	1
Ζ	7	4

- Διάσχιση της ταξινομημένης λίστας με βάση το Y
- Η βασική ιδέα είναι ότι αφαιρώ τα στοιχεία που καταστρατηγούν την φθίνουσα ταξινόμηση ως προς Y.
- Ξεκινώ τη διάσχιση από το τέλος της λίστας και συγκρίνω το τρέχον στοιχείο με το αμέσως προηγούμενο του. Ελέγχω αν η συντεταγμένη Y του τρέχοντος στοιχείου είναι μεγαλύτερη από αυτή του προηγούμενου στοιχείου.
- Αν ναι, το προηγούμενο στοιχείο αφαιρείται
- Αν όχι, η προς τα πίσω αναζήτηση συνεχίζει με το αμέσως επόμενο ζεύγος στοιχείων.
- Χρόνος: $\Theta(n)$

Άσκηση: Εύρεση k-κυρίαρχων στοιχείων πίνακα

- Έστω πίνακας A μεγέθους n, ο οποίος περιέχει πραγματικούς αριθμούς. Αναπτύξτε αλγόριθμο ο οποίος θα εντοπίζει τα στοιχεία του πίνακα A που είναι κυρίαρχα k θέσεις δεξιά και k θέσεις αριστερά τους. Για παράδειγμα το i-στό στοιχείο του πίνακα είναι κυρίαρχο αν είναι μεγαλύτερο ή ίσο με τα στοιχεία των θέσεων $j \in \{i-k, i-k+1, \dots, i+k-1, i+k\}$.

Εύρεση k-κυρίαρχων στοιχείων πίνακα

- Έστω ένας πίνακας 8 πραγματικών αριθμών.

x1	x2	x3	x4	x5	x6	x7	x8
12	9	7	11	6	10	8	3

- Με μία σάρωση του πίνακα διαπιστώνω ότι ισχύουν οι ακόλουθες σχέσεις:

x1>x2>x3<x4>x5<x6>x7>x8

- Μπορώ να βγάλω συμπεράσματα σχετικά με τα k-κυρίαρχα σημεία;

Αν ένα σημείο είναι k-κυρίαρχο τότε αποκλείεται τα γειτονικά του σημεία (k θέσεις δεξιά και k θέσεις αριστερά) να είναι και αυτά k-κυρίαρχα

Εύρεση k-κυρίαρχων στοιχείων πίνακα - 1η σάρωση

x1	x2	x3	x4	x5	x6	x7	x8
12	9	7	11	6	10	8	3

Γραμμική σάρωση του πίνακα, συγκρίνοντας το κάθε στοιχείο με τα k δεξιότερα. Όπου $k = 2$.

Εύρεση k-κυρίαρχων στοιχείων πίνακα - $2^{η}$ σάρωση

x1	x2	x3	x4	x5	x6	x7	x8
12	9	7	11	6	10	8	3

Γραμμική σάρωση του πίνακα, συγκρίνοντας το κάθε στοιχείο με τα k αριστερότερα.
Όπου $k = 2$.
Συνολικός χρόνος: $O(kn)$

Εύρεση k-κυρίαρχων στοιχείων πίνακα

- Λύση σε χρόνο $\mathrm{O}(n)$ (υπόθεση $k < n$)
- Χρήση μίας ουράς FIFO
- Αρχικά η ουρά άδεια
- Διάσχιση προς τα δεξιά
- Σε ένα ενδιάμεσο βήμα, εξετάζεται το $\mathrm{A}[i]$:
  - Αν το πρώτο στοιχείο της ουράς έχει δείκτη $< i - k$, αφαίρεση από την ουρά
  - Το $\mathrm{A}[i]$ συγκρίνεται με όλα τα στοιχεία της ουράς, και τα μικρότερα από αυτό αφαιρούνται από την ουρά.
  - Αν η ουρά αδειάσει, το $A[i]$ είναι κυρίαρχο από αριστερά.
  - Αν κάποιο στοιχείο της ουράς είναι μεγαλύτερο του $A[i]$, η διαδικασία σταματάει
  - Το $A[i]$ προστίθεται πάντα στο τέλος της ουράς
  - Χρόνος: $\Theta(n)$
  - Μήκος της ουράς $\leq k + 1$
- Συμμετρικά και η διάσχιση προς τα αριστερά

Άσκηση: Εύρεση του μέγιστου πλήθους διαστημάτων με μη κενή τομή

- Έστω n διάστημα $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, με $x_i, y_i \in \mathbb{R}, i \in 1, \dots, n$. Σχεδιάστε και αναλύστε έναν αλγόριθμο εύρεσης του μέγιστου πλήθους διαστημάτων με μη κενή τομή.

Εύρεση του μέγιστου πλήθους διαστημάτων με μη κενή τομή

Διαστήματα

(x1,y1)	(3,13)
(x2,y2)	(2,6)
(x3,y3)	(5,8)
(x4,y4)	(1,18)
(x5,y5)	(4,9)
(x6,y6)	(7,20)

Εύρεση του μέγιστου πλήθους διαστημάτων με μη κενή τομή

Διαστήματα

(x1,y1)	(3,13)
(x2,y2)	(2,6)
(x3,y3)	(5,8)
(x4,y4)	(1,18)
(x5,y5)	(4,9)
(x6,y6)	(7,20)

Εύρεση του μέγιστου πλήθους διαστημάτων με μη κενή τομή

- Ιδέα: Ταξινομούμε χωριστά τα x_i και y_i εντός χρόνου $O(nlogn)$, ώστε να προκύψουν 2 διατεταγμένες λίστες X, Y, αντίστοιχα. Κατόπιν τις σαρώνουμε, όπως θα κάναμε για να τις συγχωνεύσουμε σε μία ταξινομημένη λίστα, αυξάνοντας μία μεταβλητή c, μετρητή του μέγιστου πλήθους των τομών, κάθε φορά που «νικητής» αναδεικνύεται μέλος της X και μειώνοντας την c όποτε μέλος της Y είναι μικρότερο.

Εύρεση του μέγιστου πλήθους διαστημάτων με μη κενή τομή

Διαστήματα

(x1,y1)	(3,13)
(x2,y2)	(2,6)
(x3,y3)	(5,8)
(x4,y4)	(1,18)
(x5,y5)	(4,9)
(x6,y6)	(7,20)

Ταξινόμηση X

X	
x4	1
x2	2
x1	3
x5	4
x3	5
x6	7

Ταξινόμηση Y

Y	
y2	6
y3	8
y5	9
y1	13
y4	18
y6	20

Το μέγιστο πλήθος διαστημάτων με μη κενή τομή είναι 5, όσο και η μέγιστη τιμή που πήρε ο μετρητής C.

Άσκηση: Άθροισμα στοιχείων πίνακα

- Δοθέντος ενός ταξινομημένου πίνακα A, n διαφορετικών ακεραίων και ενός ακεραίου T, να βρεθούν δύο ακέραιοι με άθροισμα ακριβώς T.

Είσοδος (ταξινομημένη)

–20	10	20	30	35	40	60	70

$T = 60$

Άθροισμα στοιχείων πίνακα

- Ιδέα: Χρήση δύο δεικτών για τη σάρωση των στοιχείων του πίνακα A. Ο ένας δείκτης, έστω i σαρώνει τον πίνακα από $0 \rightarrow n$ και ο άλλος, έστω j από $n \rightarrow 0$.
  - Αν $\mathrm{A[i]} + \mathrm{A[j]} == T$ τότε επιστρέφει τα $A[i], A[j]$.
  - Αν $\mathrm{A[i]} + \mathrm{A[j]} < T$ τότε ο δείκτης i μετακινείται μια θέση δεξιά (δείχνει στο επόμενο στοιχείο του πίνακα)
  - Αν $\mathrm{A[i]} + \mathrm{A[j]} > T$ τότε ο δείκτης j μετακινείται μια θέση αριστερά (δείχνει στο προηγούμενο στοιχείο του πίνακα)

Είσοδος (ταξινομημένη)

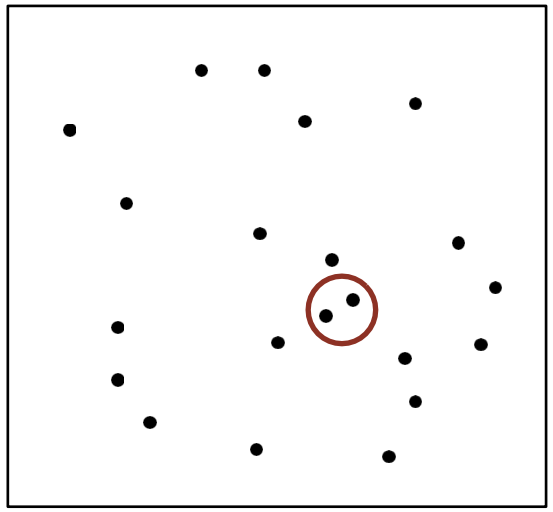
–20	10	20	30	35	40	60	70
		↑ *i*			↑ *j*		

T = 60

Χρόνος: O(n)

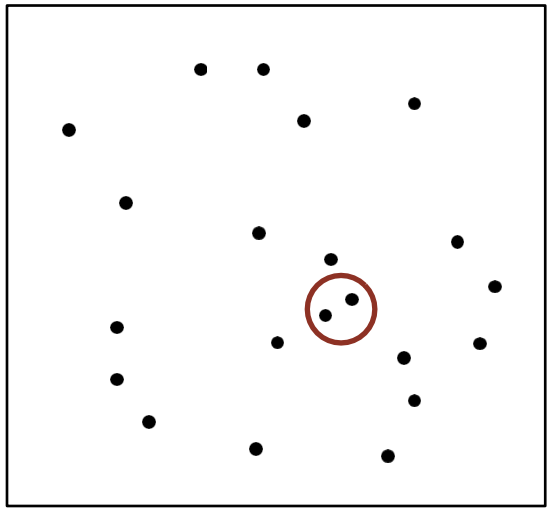
Άσκηση : Πλησιέστερο ζεύγος σημείων

- 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

- 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.

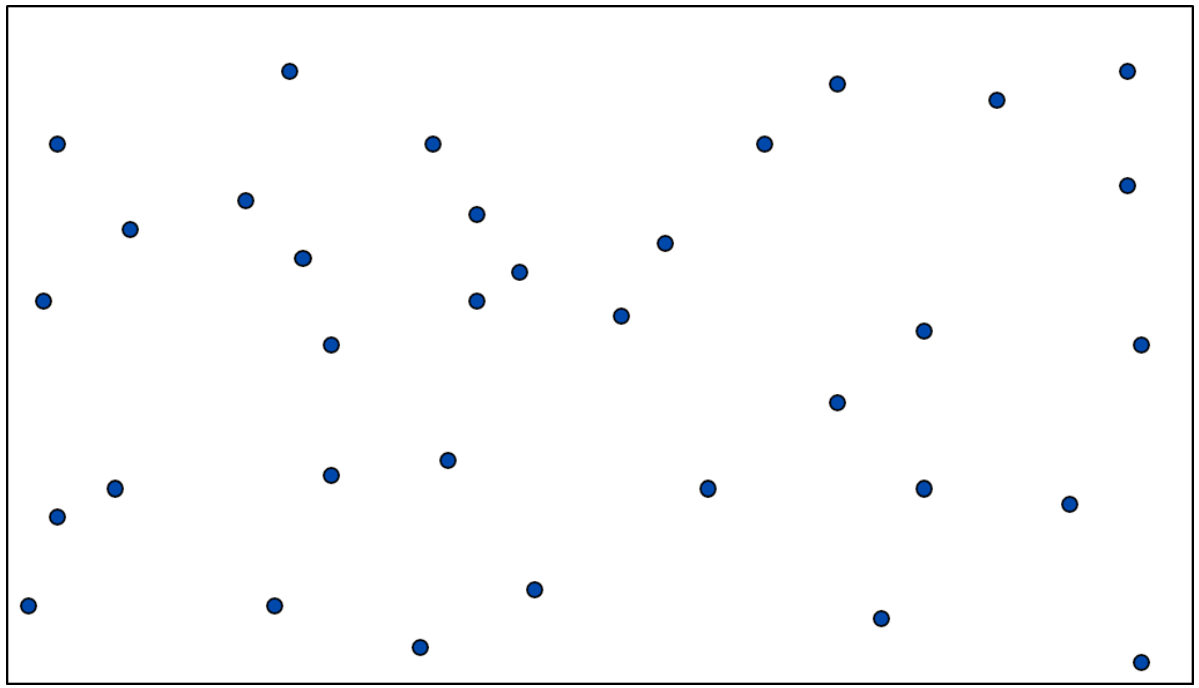


Πλησιέστερο ζεύγος σημείων: πρώτη προσπάθεια

Λύση βάσει ταξινόμησης.

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

Ταξινόμησε ως προς την y-συντεταγμένη και θεώρησε κοντινά σημεία

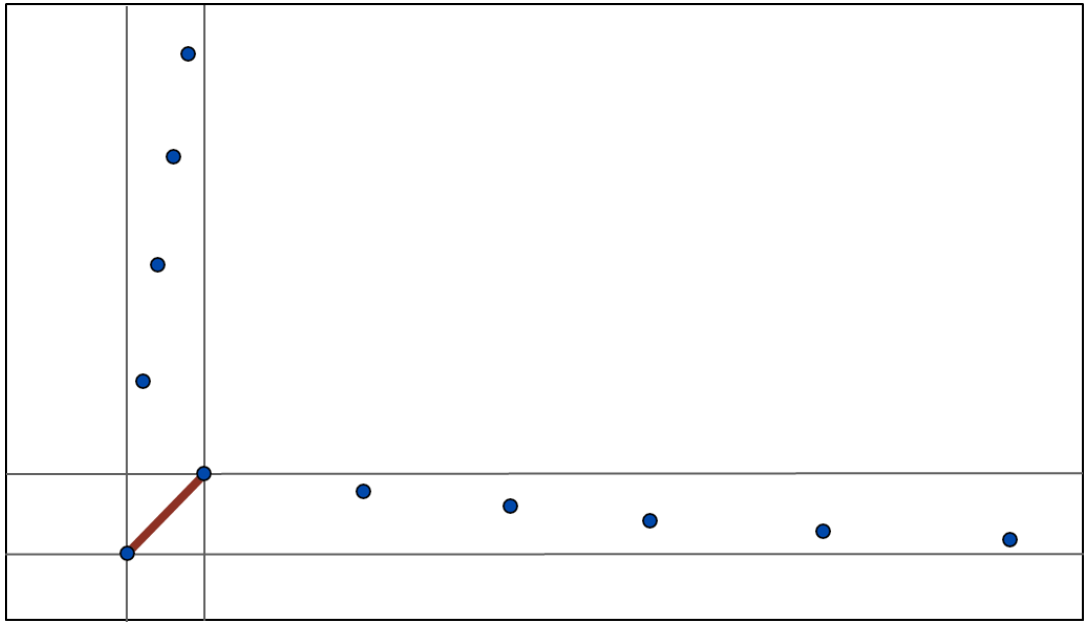


Πλησιέστερο ζεύγος σημείων: πρώτη προσπάθεια

Λύση βάσει ταξινόμησης.

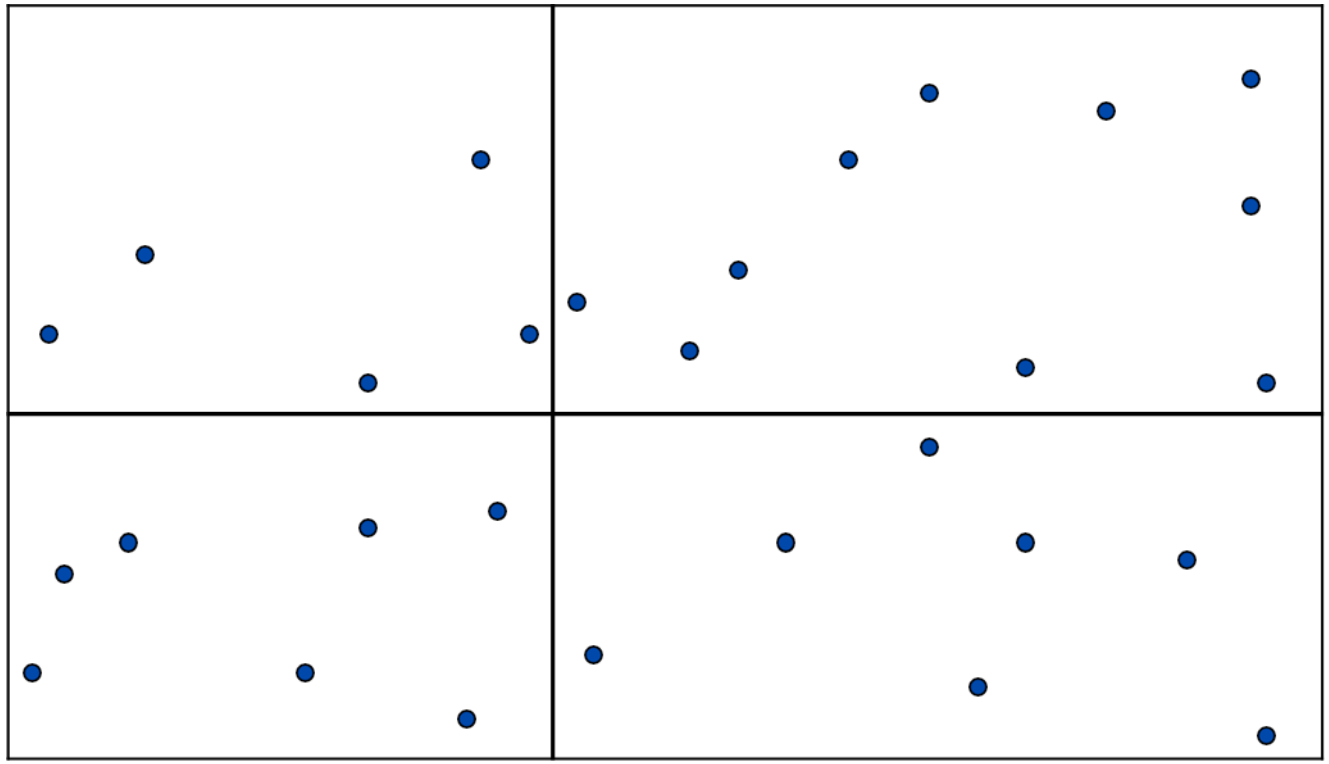
Ταξινόμησε ως προς τη x-συντεταγμένη και θεώρησε κοντινά σημεία.

Ταξινόμησε ως προς την y-συντεταγμένη και θεώρησε κοντινά σημεία



Πλησιέστερο ζεύγος σημείων: δεύτερη προσπάθεια

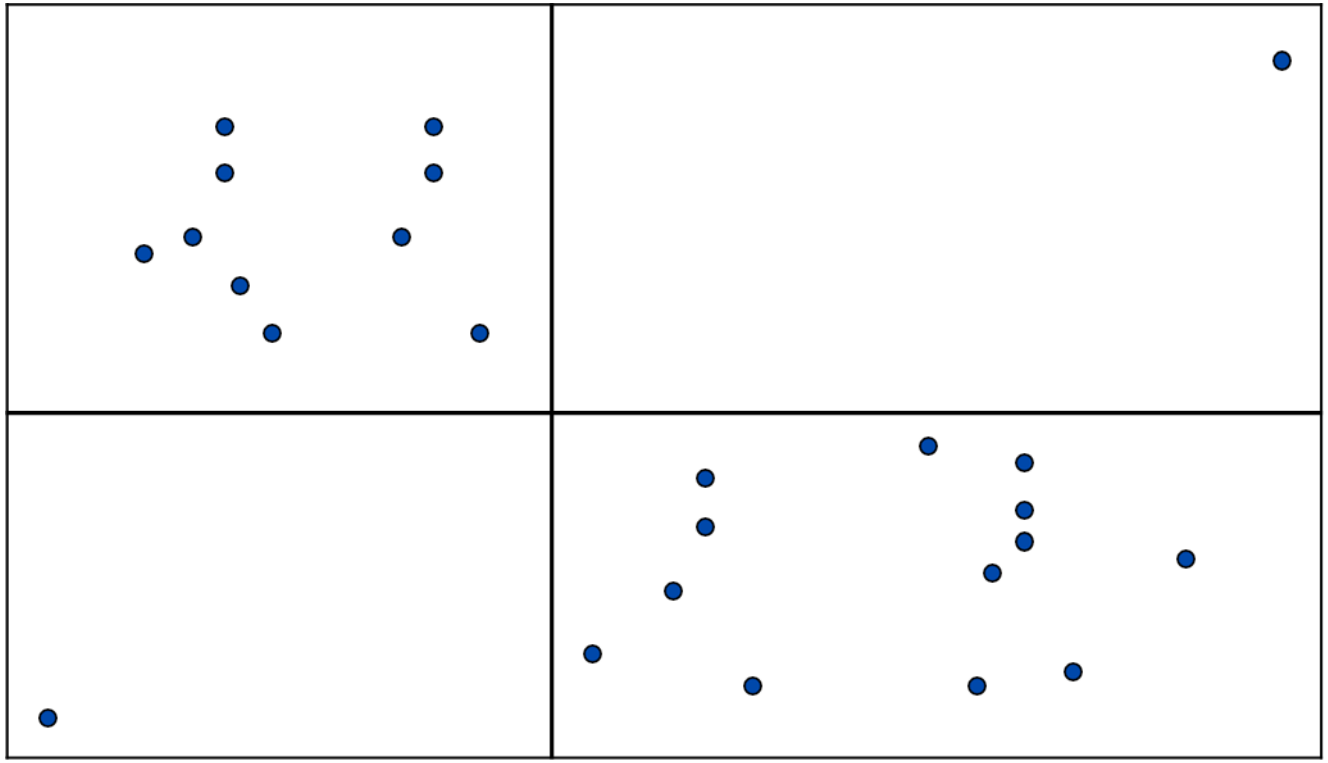
Διαίρεσε. Υποδιαίρεσε την περιοχή σε 4 τεταρτημόρια.



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

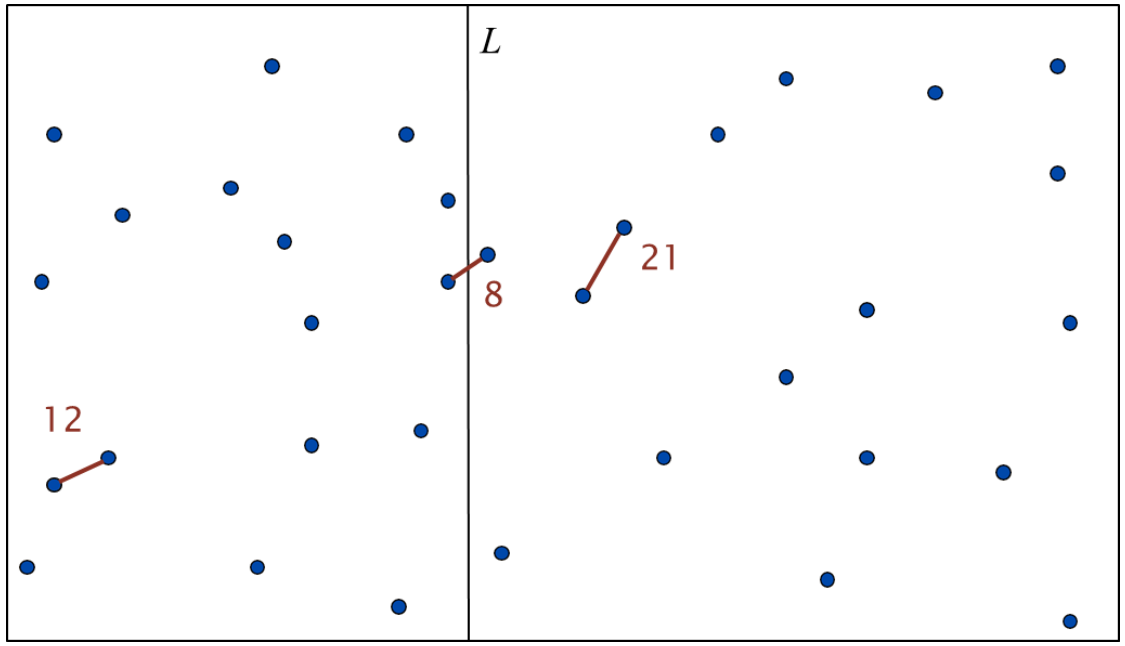
Διαίρεσε: Σχεδίασε μία κάθετη γραμμή L έτσι ώστε $n/2$ σημεία σε κάθε πλευρά.

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.

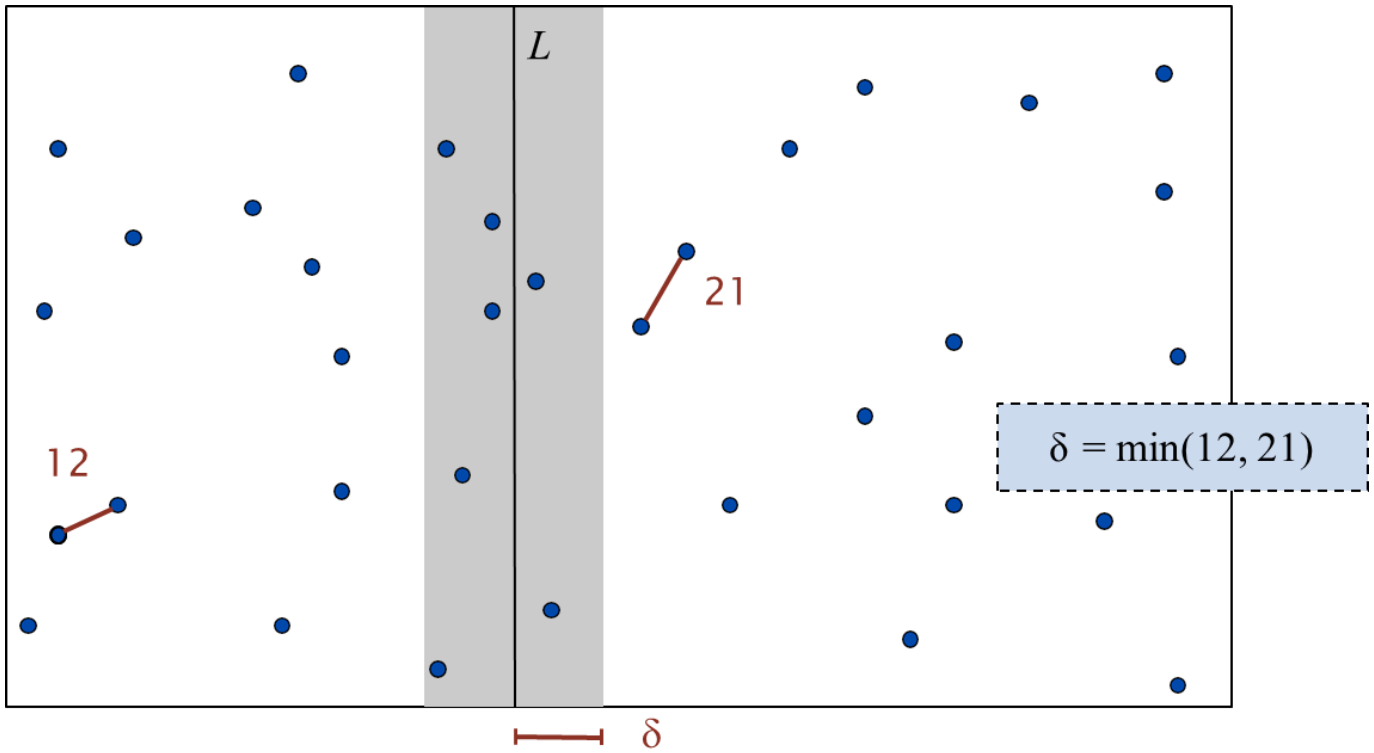
Φαίνεται ότι απαιτεί $\Theta(n^2)$



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 είναι μικρότερη από δ.

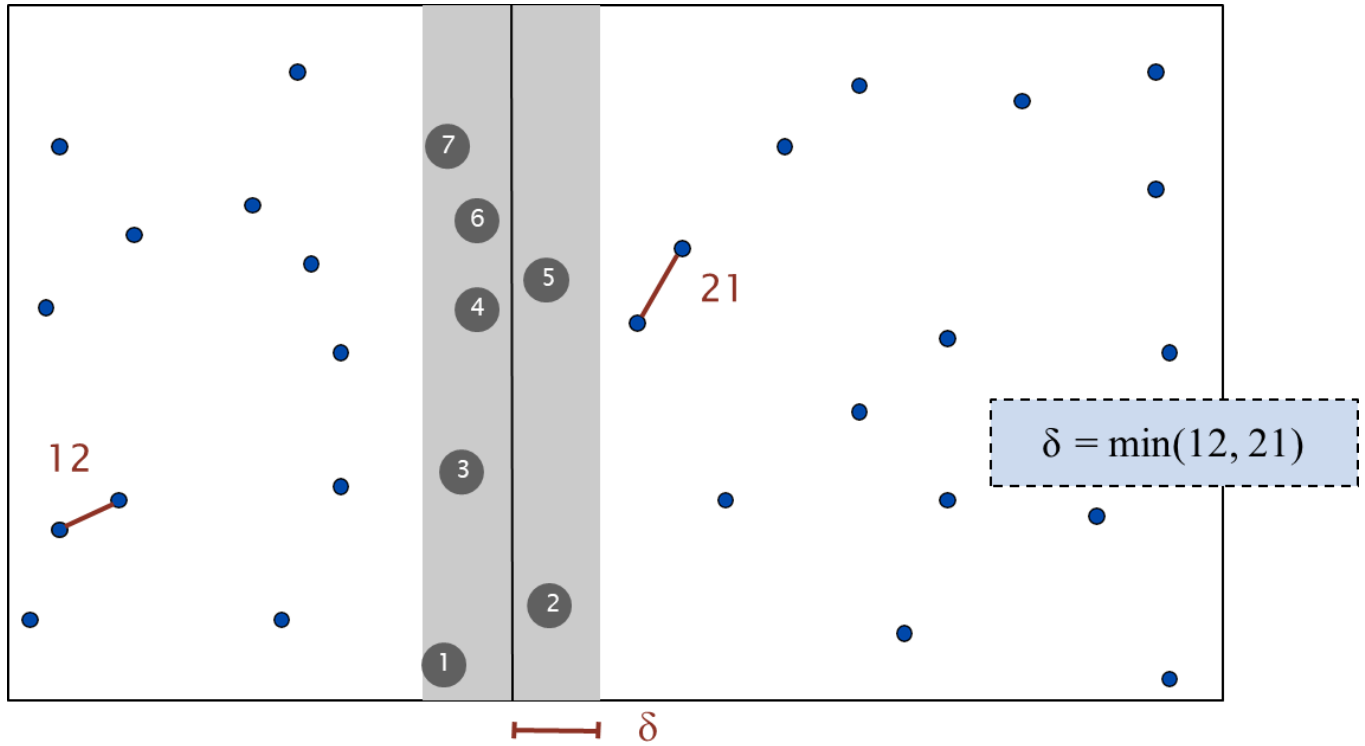
Παρατήρηση: αρκεί να θεωρήσουμε μόνο εκείνα τα σημεία εντός απόστασης δ από τη γραμμή L.



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

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;

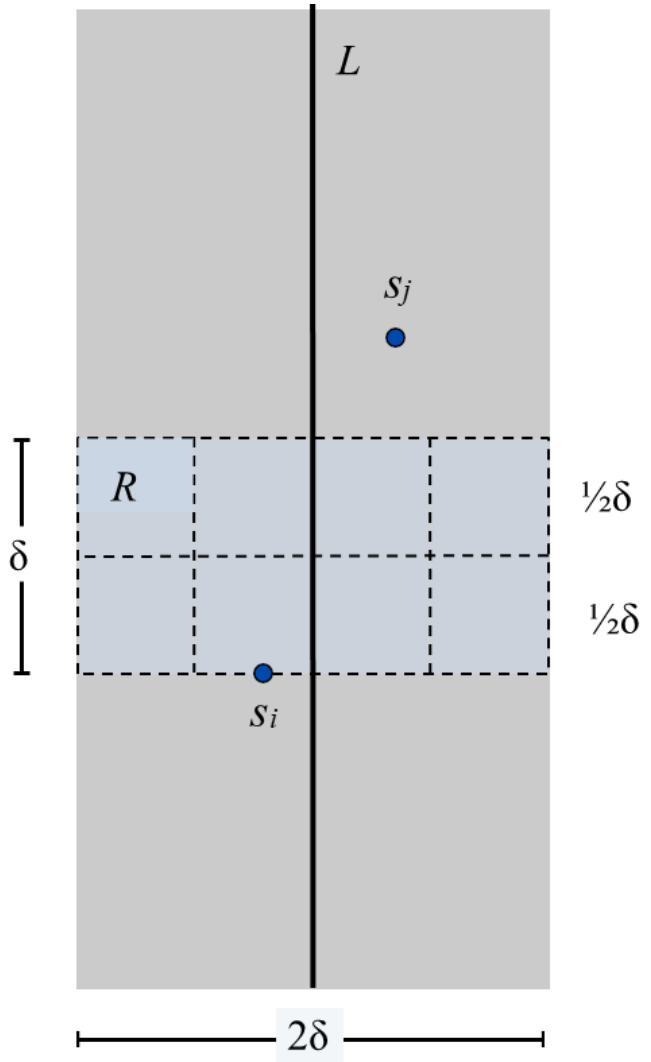
Ορισμός: Έστω s_i το σημείο στη λωρίδα πλάτους 2δ, με την $i^{οστη}$ μικρότερη y-συντεταγμένη.

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

Απόδειξη:

- Θεώρησε το $2\delta \times \delta$ ορθογώνιο R σε μία λωρίδα της οποίας η ελάχιστη y-συντεταγμένη είναι η y-συντεταγμένη του s_i.
- Απόσταση μεταξύ s_i και κάθε σημείου s_j
  - Πάνω από το $R \geq \delta$.
- Υποδιαίρεσε το R σε 8 τετράγωνα.
- Το πολύ 1 σημείο ανά τετράγωνο.
- Το πολύ 7 άλλα σημεία μπορεί να είναι στο R.

Διάμετρος:
$$\frac{\delta}{\sqrt{2}} < \delta$$



Πλησιέστερο ζεύγος σημείων: Αλγόριθμος διαίρει και βασίλευε

Πλησιέστερο-Ζεύγος $(p_1, p_2, \dots, p_n)$

1. Υπολόγισε το κατακόρυφο ευθύγραμμο τμήμα L που χωρίζει σε τα σημεία σε δύο ίδιας πληθικότητας σύνολα ← $O(n)$
2. $\delta_1 \leftarrow$ *Πλησιέστερο-Ζεύγος(σημεία στο αριστερό επίπεδο)* ← $T(n/2)$
3. $\delta_2 \leftarrow$ *Πλησιέστερο-Ζεύγος(σημεία στο δεξί επίπεδο)* ← $T(n/2)$
4. $\delta \leftarrow \min\{\delta_1, \delta_2\}$
5. Διέγραψε όλα τα σημεία που βρίσκονται απόσταση $> \delta$ από το L ← $O(n)$
6. Ταξινόμησε τα υπόλοιπα σημεία βάσει της y-συντεταγμένης ← $O(nlogn)$
7. Διέτρεξε τα σημεία βάσει της διάταξης της y-συντεταγμένης και υπολόγισε την απόστασή τους με τους 7 επόμενους γείτονές τους. Αν κάποιο έχει απόσταση $< \delta \Rightarrow$ ενημέρωσε την απόσταση δ. ← $O(n)$
8. Επέστρεψε τη απόσταση δ

Ποια είναι η λύση της ακόλουθης αναδρομής;

$$T(n) = \begin{cases} \Theta(1) & n = 1 \\ T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + T\left(\left\lceil \frac{n}{2} \right\rceil\right) + \Theta(nlogn) & n > 1 \end{cases}$$

A. $T(n) = \Theta(n)$
B. $T(n) = \Theta(n\ logn)$
C. $T(n) = \Theta(n\ log^2 n)$
D. $T(n) = \Theta(n^2)$

Άσκηση: Πλειοψηφούν στοιχείο πίνακα

Έστω πίνακας A, μεγέθους n. Ένα στοιχείο του πίνακα A καλείται πλειοψηφούν στοιχείο (majority element) εάν εμφανίζεται περισσότερες από $n/2$ φορές.

Ακολουθώντας την τεχνική του διαίρει και βασίλευε, σχεδιάστε και αναλύστε έναν αλγόριθμο, ο οποίος θα εντοπίζει το στοιχείο πλειοψηφίας (majority element) σε χρόνο $O(nlogn)$.

Σημείωση: Έχετε στη διάθεσή σας μόνο την πράξη της ισότητας και δεν μπορείτε να χρησιμοποιήσετε αλγόριθμο ταξινόμησης

Πλειοψηφούν στοιχείο πίνακα

- Έστω πίνακας $\mathbf{A}, \boldsymbol{n}$ στοιχείων. Ένα στοιχείο του καλείται στοιχείο πλειοψηφίας (majority element) αν εμφανίζεται περισσότερες από $\boldsymbol{n/2}$ φορές.

Ιδέα: Θα χρησιμοποιήσουμε την τεχνική του διαίρει και βασίλευε.

Εντοπίζουμε αναδρομικώς στο αριστερό μισό και στο δεξί μισό του πίνακα τα στοιχεία πλειοψηφίας, έστω a, b, αντιστοίχως. Κατόπιν, μετράμε, με γραμμική σάρωση στο εύρος όλου του πίνακα το πλήθος εμφανίσεων των a, b έστω c, d, αντίστοιχα.

Αν $c > d$ και $c > n/2$, τότε το a είναι το αναζητούμενο στοιχείο.

Αλλιώς, αν $c < d$ και $d > n/2$, τότε το b είναι το αναζητούμενο στοιχείο.

Διαφορετικά, δεν υπάρχει στοιχείο πλειοψηφίας στον A.

Πλειοψηφούν στοιχείο πίνακα

- Πολυπλοκότητα αλγορίθμου:

Η χρονική πολυπλοκότητα του αλγορίθμου που περιγράφηκε, δίδεται από την ακόλουθη αναδρομική εξίσωση:

$$\mathrm{T(n)} = 2\mathrm{T}\left(\frac{\mathrm{n}}{2}\right) + \mathrm{kn}, \mathrm{k}\ \sigma\tau\alpha\theta\epsilon\rho\acute{\alpha}$$

Πολυπλοκότητα εύρεσης λύσης σε πίνακα μεγέθους n

Πολυπλοκότητα εύρεσης λύσης σε πίνακα μεγέθους n/2

Κόστος συνένωσης των 2 λύσεων

Τελική πολυπλοκότητα αλγορίθμου:

$$\mathrm{T(n)} = \mathrm{logn} \cdot kn = nlogn$$

Θα διασπάσω το αρχικό πρόβλημα μεγέθους n, μέχρι να φτάσω σε μέγεθος προβλήματος όπου το πρόβλημα λύνεται τετριμμένα. Επομένως θα χρειαστεί να εφαρμόσω $logn$ φορές τη λειτουργία της συνένωσης 2 επιμέρους λύσεων.

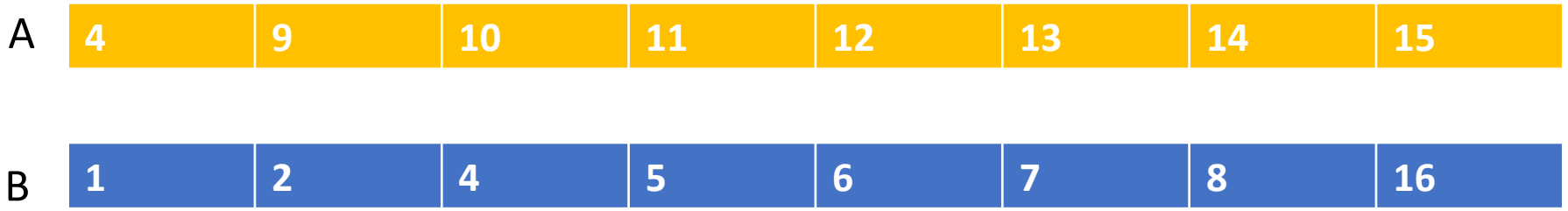
Άσκηση: Εύρεση μεσαίου στοιχείου δύο ταξινομημένων πινάκων

Έστω δύο ταξινομημένοι πίνακες A, B μεγέθους n έκαστος. Περιγράψτε και αναλύστε έναν αλγόριθμο, ο οποίος εντοπίζει το μεσαίο, από άποψη διατάξεως, στοιχείο των δύο πινάκων.

Εύρεση μεσαίου στοιχείου δύο ταξινομημένων πινάκων

- Έστω 2 ταξινομημένοι πίνακες A, B μεγέθους n έκαστος.

- Παράδειγμα:



Ζητούμενο: το μεσαίο από άποψη συνολικής διάταξης στοιχείο

1 2 4 4 5 6 7 8 9 10 11 12 13 14 15 16

Εύρεση μεσαίου στοιχείου δύο ταξινομημένων πινάκων

- Προτεινόμενη λύση:

Εφόσον οι πίνακες είναι ταξινομημένοι, ξεκινούμε συγκρίνοντας μεταξύ τους τα στοιχεία $\mathrm{A}[n/2], B[n/2]$ της μεσαίας θέσης.

❖Αν είναι ίσα, τότε έχουμε εντοπίσει το μεσαίο στοιχείο.

❖Αν $\mathrm{A}[n/2] > B[n/2]$ τότε υποχρεωτικά το μεσαίο στοιχείο:

α) δεν μπορεί να ανήκει στον A και να είναι μεγαλύτερο του $\mathrm{A}[n/2]$

β) δεν μπορεί να ανήκει στον B και να είναι μικρότερο ή ίσο του $\mathrm{B}[n/2]$

❖Αν $\mathrm{A}[n/2] < B[n/2]$ τότε υποχρεωτικά το μεσαίο στοιχείο:

α) δεν μπορεί να ανήκει στον A και να είναι μικρότερο ή ίσο του $\mathrm{A}[n/2]$

β) δεν μπορεί να ανήκει στον B και να είναι μεγαλύτερο του $\mathrm{B}[n/2]$

Εύρεση μεσαίου στοιχείου δύο ταξινομημένων πινάκων

- Η χρονική πολυπλοκότητα του αλγορίθμου ακολουθεί την εξής αναδρομική σχέση:

$$T(2n) = T(n) + c$$

Συνολική πολυπλοκότητα: $O(logn)$

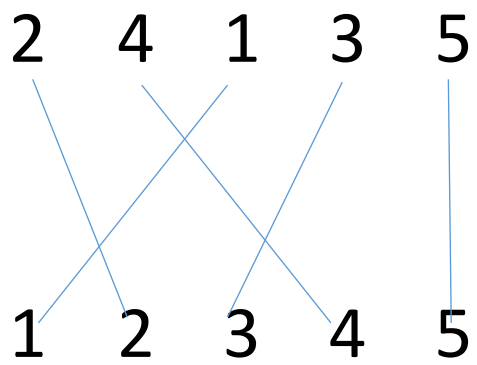
Άσκηση: Εύρεση αναστροφών

Έστω το πρόβλημα της εύρεσης των ατόμων με κοινά ενδιαφέροντα-προτιμήσεις. Βασικό κομμάτι του προβλήματος είναι η σύγκριση δύο κατατάξεων (rankings). Έστω για παράδειγμα ότι κατατάσσουμε με βάση την προτίμηση μας ένα σύνολο n ταινιών και το ζητούμενο είναι να βρούμε άλλους χρήστες του διαδικτύου με παρόμοιες προτιμήσεις με τις δικές μας. Υπάρχει κάποια μετρική που να αξιολογεί την ομοιότητα δύο κατατάξεων από δύο διαφορετικούς χρήστες;

Ένας τρόπος να εκτιμήσουμε την ομοιότητα είναι να αριθμήσουμε τις ταινίες με αύξουσα σειρά με βάση την προτίμηση του άλλου χρήστη. Ο αριθμός των αναστροφών (inversions) ποσοτικοποιεί το βαθμό ομοιότητας των δυο κατατάξεων. Αν ισχύει $i < j$ και $A[i] > A[j]$, τότε το ζεύγος των (i, j) ονομάζεται ανάστροφο ζεύγος.

Προτείνετε και αναλύστε έναν αλγόριθμο που θα εκτιμά το βαθμό ομοιότητας δύο κατατάξεων μεγέθους n σε χρόνο $O(n \log n)$.

Εύρεση αναστροφών



Ανάστροφα ζεύγη (i, j): $i < j$ και $A[i] > A[j]$, με $i, j \in [1, n]$

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

Γραφικός τρόπος εύρεσης του πλήθους των ανάστροφων ζευγαριών: εύρεση του πλήθους των τομών των ευθυγράμμων τμημάτων

Εύρεση αναστροφών

- $0 \leq$ πλήθος ανάστροφων ζευγών $\leq \binom{\mathrm{n}}{2}$

- Απλός αλγόριθμος: Εξετάζω όλα τα πιθανά ζευγάρια προκειμένου να δω αν είναι ανάστροφα $\rightarrow \binom{\mathrm{n}}{2} = \frac{n!}{2!(n-2)!} = \mathrm{O}(n^2)$

- Θέλουμε να χαμηλώσουμε την πολυπλοκότητα σε $\mathrm{O}(n\ logn)$ γεγονός το οποίο συνεπάγεται ότι δεν πρέπει καν να εξετάσουμε όλα τα πιθανά ζευγάρια → Τεχνική Διαίρει και Βασίλευε

Εύρεση αναστροφών

- Έστω ότι διασπούμε την αρχική λίστα μεγέθους n, σε 2 υπολίστες μεγέθους $\mathrm{m} = \lceil n/2 \rceil$
- Προκύπτουν οι υπολίστες:

$\mathrm{A} = a_1, \dots, a_m$ και $\mathrm{B} = a_{m+1}, \dots, a_n$

- Έστω ότι γνωρίζω το πλήθος των ανάστροφων ζευγών της κάθε υπολίστας.
- Ζητούμενο λοιπόν είναι να βρω τα ανάστροφα ζεύγη που σχηματίζονται μεταξύ στοιχείων της πρώτης και δεύτερης υπολίστας.
- Αν $a \in \mathrm{A}$ στοιχείο της πρώτης υπολίστας και $b \in \mathrm{B}$ στοιχείο της δεύτερης υπολίστας:

Το ζεύγος (a, b) είναι ανάστροφο αν ισχύει $a > b$, δεδομένου ότι
η $\theta\acute{\epsilon}\sigma\eta$ του α είναι μικρότερη του b.

- Ο υπολογισμός των ανάστροφων ζευγών μεταξύ στοιχείων των υπολιστών A και B πρέπει να γίνει σε χρόνο $O(n)$.

Εύρεση αναστροφών

❖Αλγόριθμος συνένωσης = Αλγόριθμος Merge and Count

- Υποθέστε πως οι υπολίστες Α, Β είναι ταξινομημένες.
- Ζητούμενο είναι να δημιουργήσουμε μία συνολική ταξινομημένη λίστα, μετρώντας παράλληλα τα ανάστροφα ζεύγη (a, b) με $a \in \mathrm{A}, b \in \mathrm{B}$ και $a > b$
- Εφαρμόζω τον αλγόριθμο της συνένωσης 2 ταξινομημένων λιστών διατηρώντας παράλληλα ένα μετρητή counter που μετρά το πλήθος των ανάστροφων ζευγών.
  - Κάθε φορά που εισάγουμε στοιχείο της υπολίστας Α στην τελική ταξινομημένη λίστα δεν δημιουργείται πρόβλημα (δεν συναντάμε ανάστροφο ζεύγος, γιατί;)
  - Αν όμως εισάγουμε στην τελική ταξινομημένη λίστα στοιχείο b της υπολίστας Β αυτό συνεπάγεται ότι το b είναι μικρότερο από όλα τα εναπομείναντα στοιχεία της υπολίστας Α που δεν έχουν εισαχθεί στην τελική ταξινόμηση.

  Επομένως πρέπει να αυξήσω το μετρητή counter όσο και το πλήθος των στοιχείων που παραμένουν στη λίστα Α.

❖Πολυπλοκότητα αλγορίθμου Merge and Count: $O(n)$

❖Συνολική πολυπλοκότητα αλγορίθμου εύρεσης πλήθους ανάστροφων ζευγών: $\mathrm{O}(n\ logn)$

Αλγόριθμος Merge & Count - Παράδειγμα

Πίνακας Α

1	2	3	4
2	3	5	7

Πίνακας Β

5	6	7	8
1	4	6	8

Αλγόριθμος Merge & Count - Παράδειγμα

Πίνακας Α

1	2	3	4
2	3	5	7

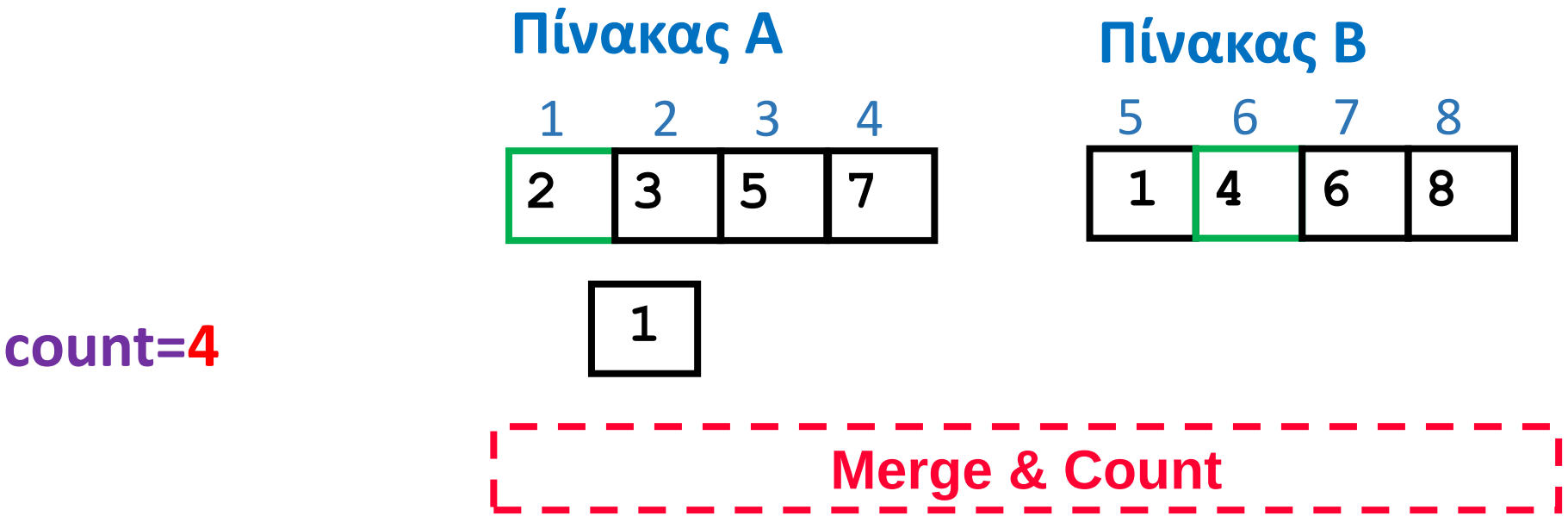
Πίνακας Β

5	6	7	8
1	4	6	8

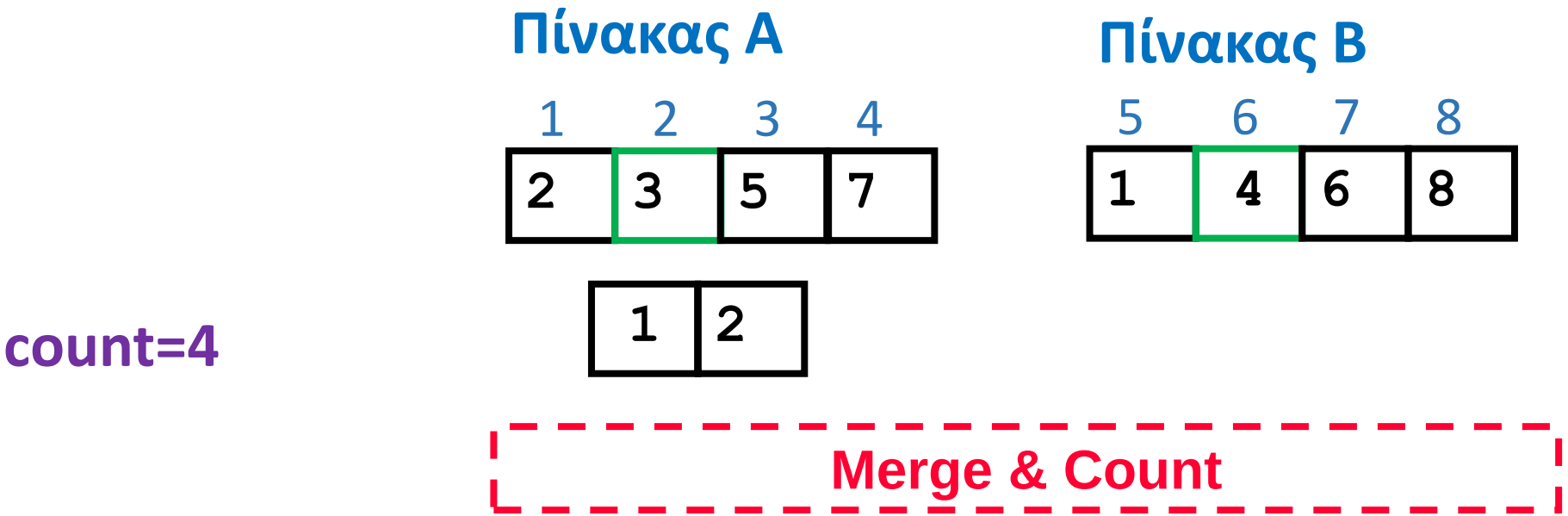
count=0

Merge & Count

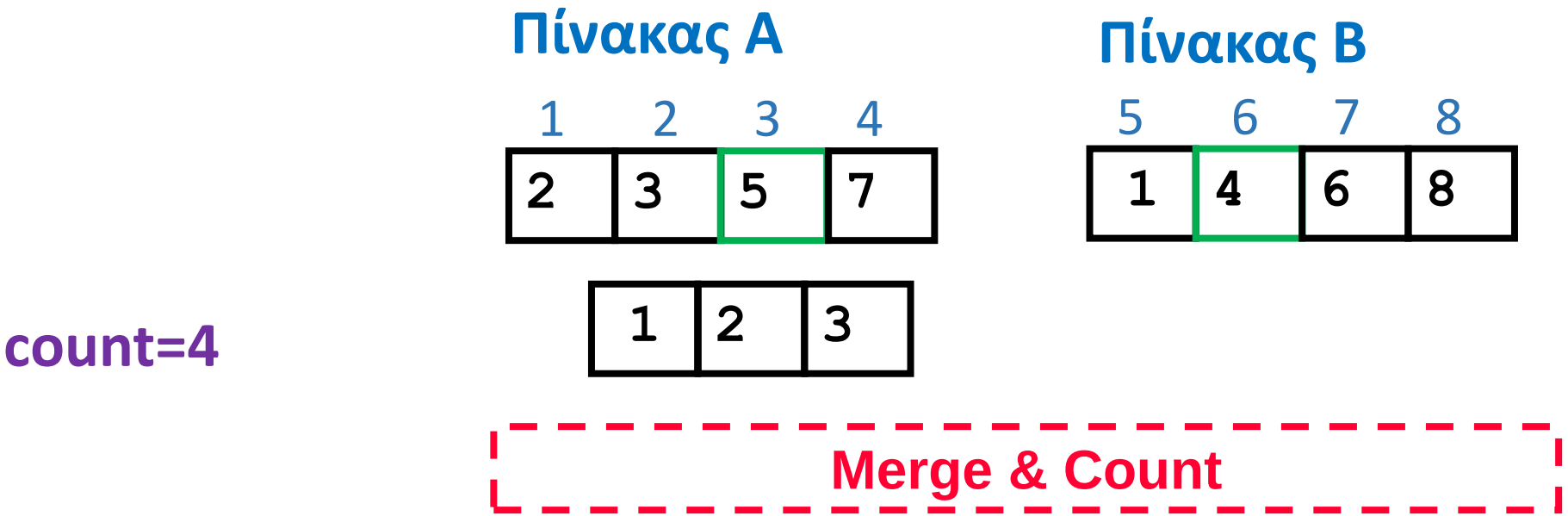
Αλγόριθμος Merge & Count - Παράδειγμα



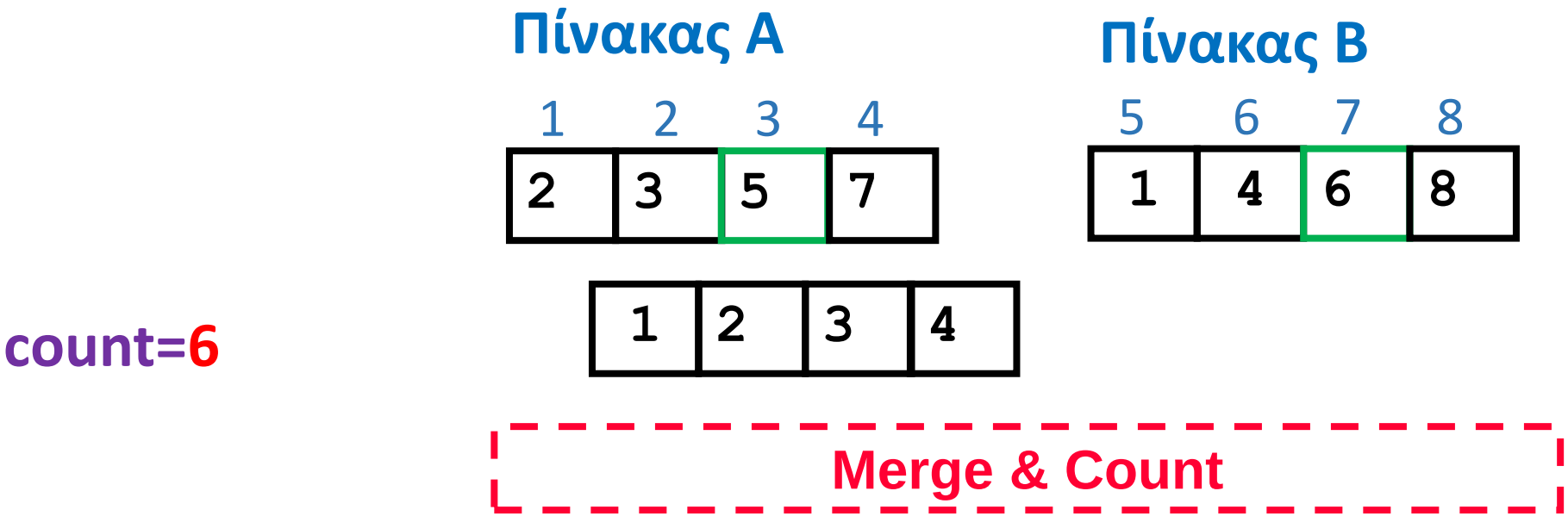
Αλγόριθμος Merge & Count - Παράδειγμα



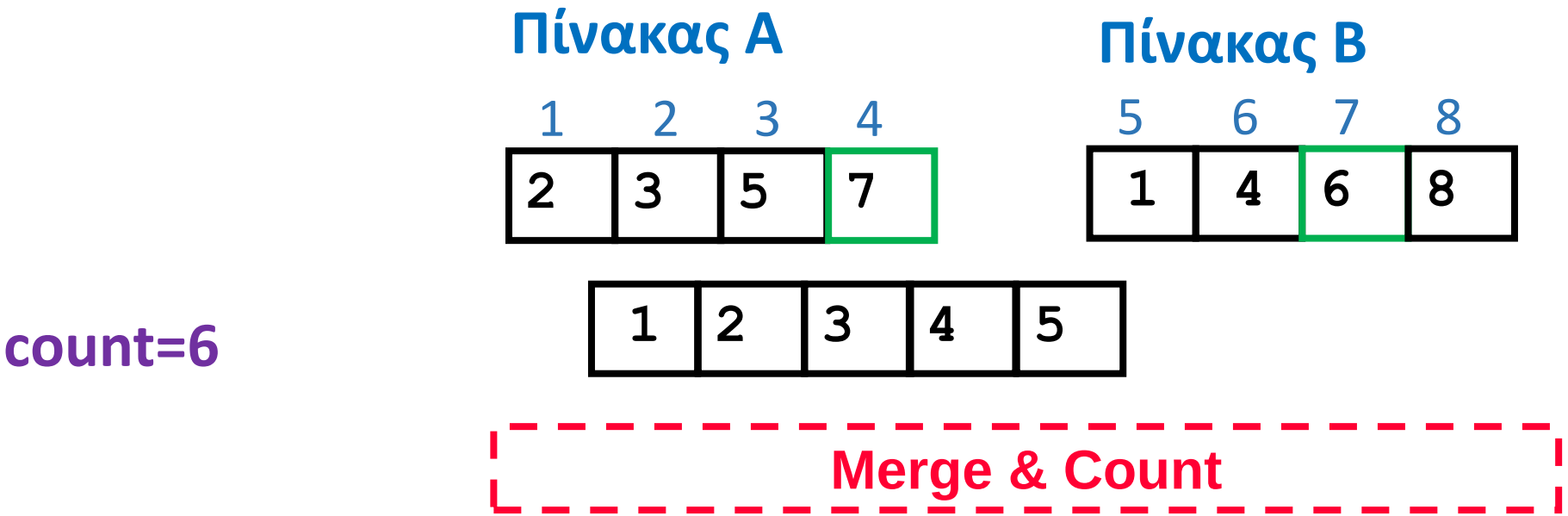
Αλγόριθμος Merge & Count - Παράδειγμα



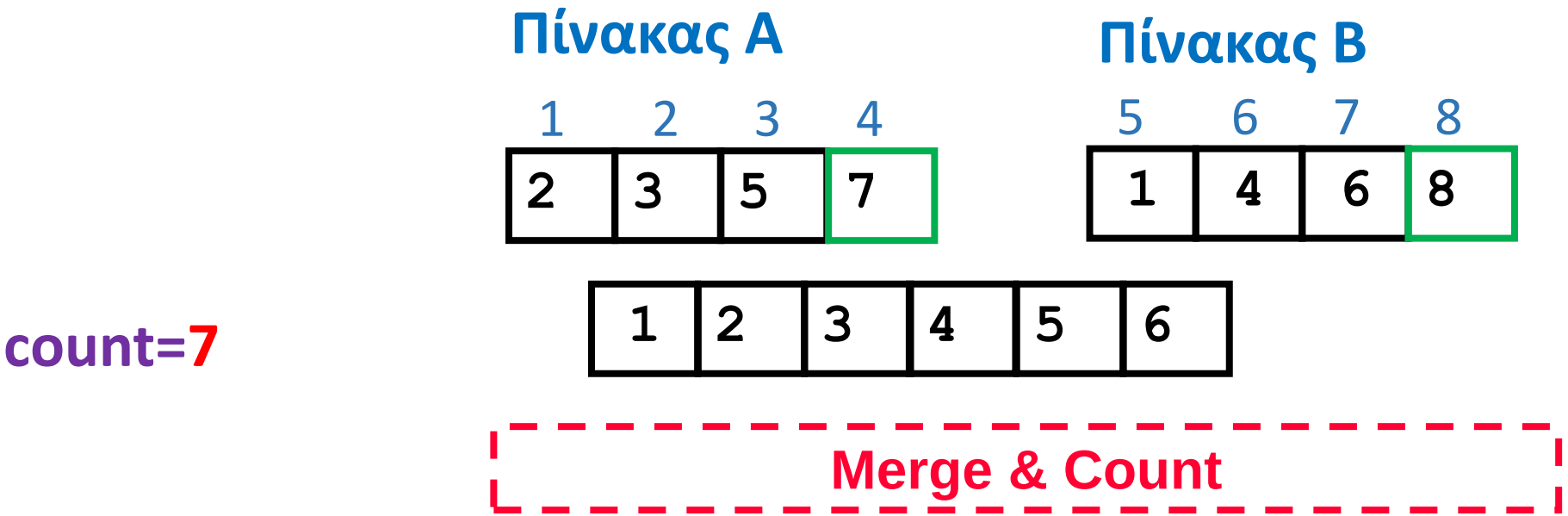
Αλγόριθμος Merge & Count - Παράδειγμα



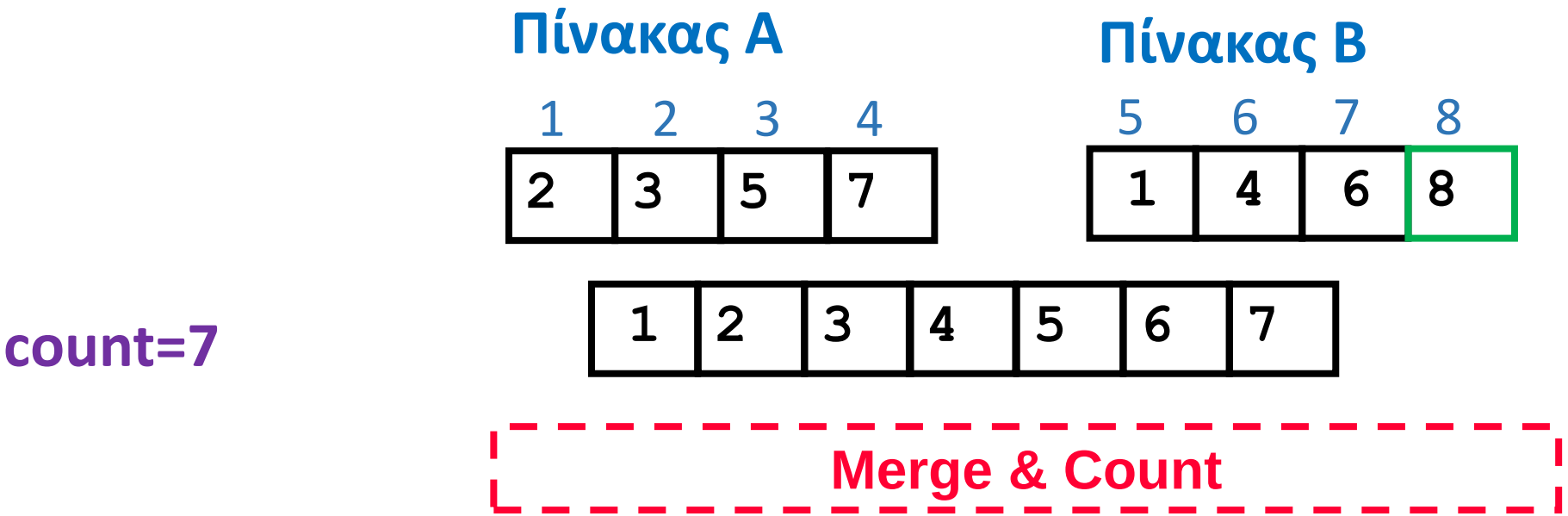
Αλγόριθμος Merge & Count - Παράδειγμα



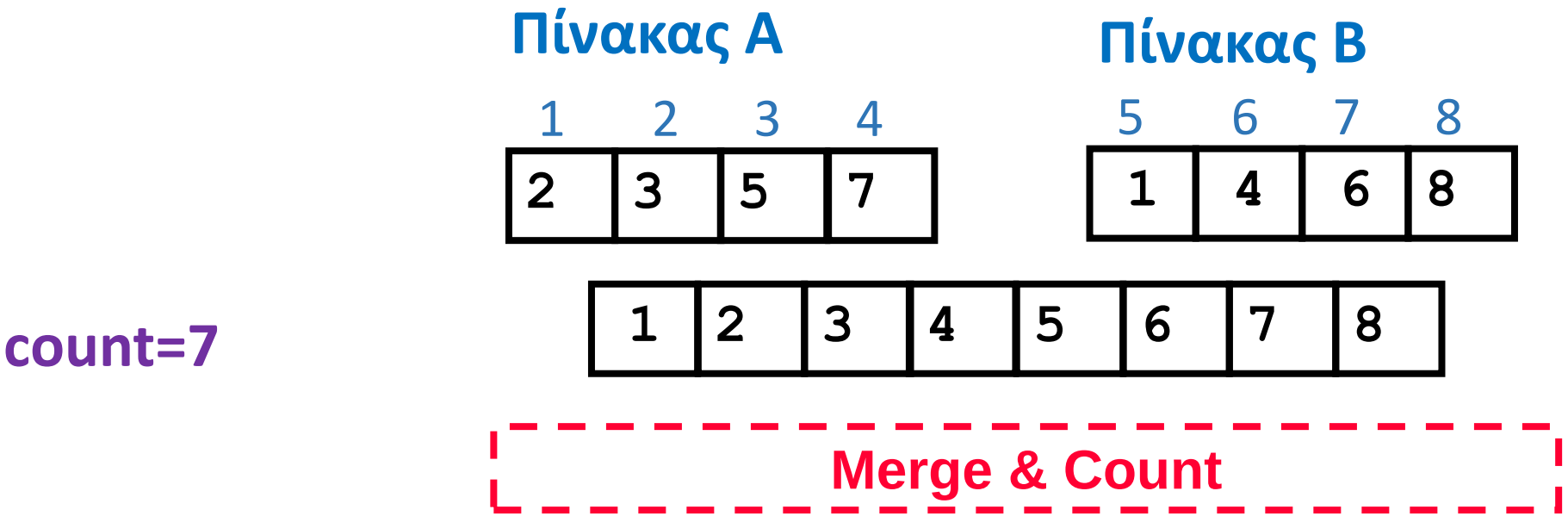
Αλγόριθμος Merge & Count - Παράδειγμα



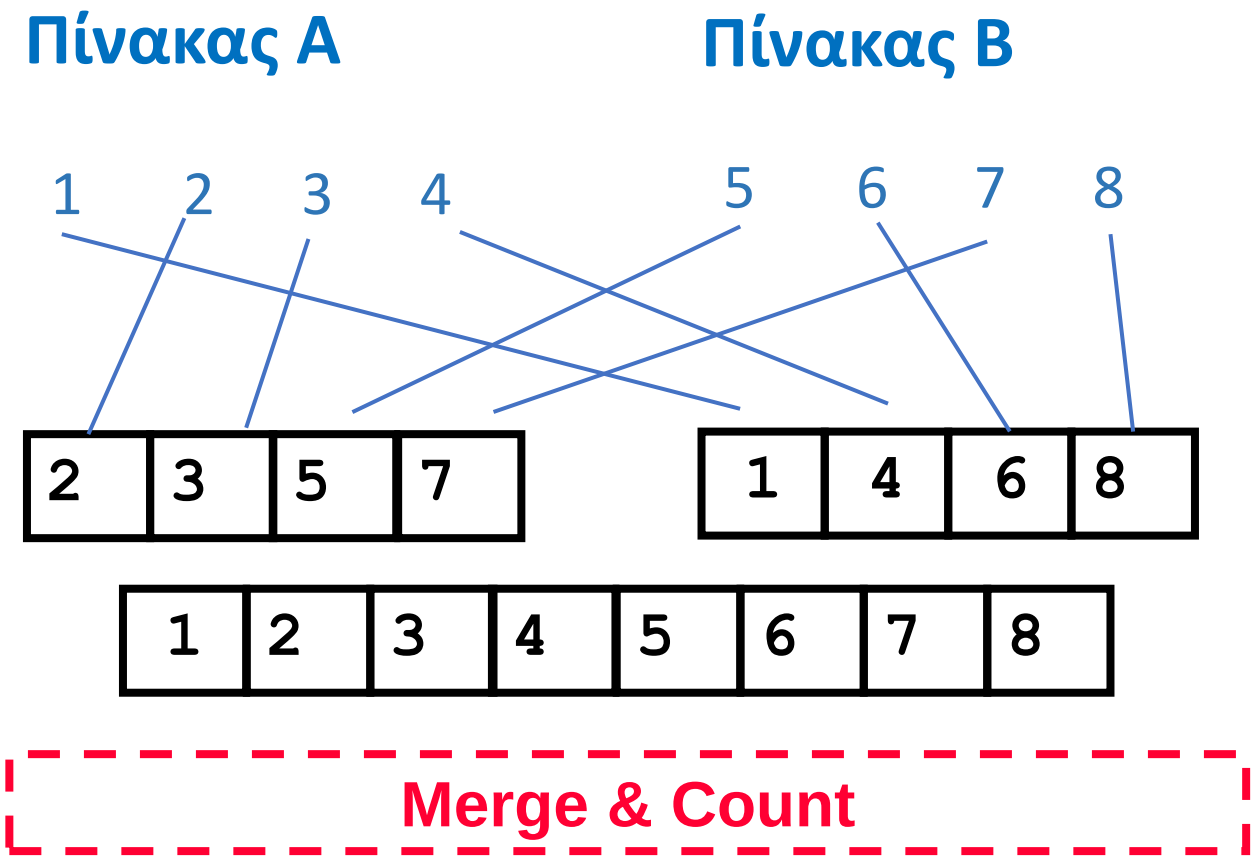
Αλγόριθμος Merge & Count - Παράδειγμα



Αλγόριθμος Merge & Count - Παράδειγμα



Αλγόριθμος Merge & Count - Παράδειγμα



count=7

Άσκηση: Γραμμική αναζήτηση σε λίστα

Έστω το πρόβλημα αναζήτησης στοιχείου σε λίστα Α και γνωρίζουμε εκ των προτέρων την πιθανότητα αναζήτησης p_i ενός στοιχείου α_i που βρίσκεται στη θέση l_i της λίστας. Θα μπορούσαμε με κάποια αναδιάταξη της λίστας να μειώσουμε την πολυπλοκότητα της μέσης περίπτωσης του αλγορίθμου της γραμμικής αναζήτησης;

Στην περίπτωση που δεν γνωρίζουμε την πιθανότητα αναζήτησης των στοιχείων, υπάρχει κάποια τεχνική που να βελτιώνει την επίδοση του αλγορίθμου γραμμικής αναζήτησης στη μέση περίπτωση;

Αναλύστε και τεκμηριώστε τις απαντήσεις σας

Γραμμική αναζήτηση σε λίστα

- Έστω ότι ένα στοιχείο α_i της λίστας βρίσκεται στη θέση l_i. Ποιο είναι το κόστος αναζήτησης του στοιχείου α_i;

- Έστω ότι είναι γνωστή η πιθανότητα αναζήτησης p_i κάθε στοιχείου α_i της λίστας, με $1 \leq i \leq n, p_i \geq 0$ $\kappa\alpha\iota$ $\sum_i p_i = 1$.

- Αναμενόμενο ή Μέσο κόστος αναζήτησης ενός στοιχείου:

$$\sum_i p_i l_i$$

Γραμμική αναζήτηση σε λίστα

- Βελτίωση μέσου χρόνου αναζήτησης στοιχείου, αν ταξινομήσουμε τη λίστα με βάση την πιθανότητα αναζήτησης (φθίνουσα σειρά).

- Απόδειξη:

Έστω δύο στοιχεία α_i, α_jτης λίστας, με πιθανότητα αναζήτησης p_i, p_j και βρίσκονται στη θέση l_i, l_j αντίστοιχα.

Έστω ότι ισχύει $p_i > p_j$ και $l_i > l_j$.

Εναλλάσσοντας τις θέσεις των στοιχείων α_i, α_jτο κόστος αναζήτησης μεταβάλλεται ως εξής:

Μεταβολή κόστους αναζήτησης=Τελικό κόστος-Αρχικό κόστος=

$$= -(p_i l_i + p_j l_j) + (p_i l_j + p_j l_i) = -(p_i - p_j)(l_i - l_j) < 0$$

Γραμμική αναζήτηση σε λίστα

- Βελτιστοποίηση μέσου κόστους αναζήτησης:

Διατάσσω και αριθμώ τα n στοιχεία με βάση τη φθίνουσα σειρά της πιθανότητας εμφάνισης:

$$p_1 \geq p_2 \geq \cdots \geq p_n \text{ και } l_i = i$$

$$\mathbf{Opt} = \sum_i \boldsymbol{p_i l_i} = \sum_i \boldsymbol{p_i i}$$

Γραμμική αναζήτηση σε λίστα

- Έστω ότι δεν γνωρίζουμε τίποτα σχετικά με τις πιθανότητες αναζήτησης των στοιχείων της λίστας.

Ιδέα: Ένα συχνά αναζητούμενο στοιχείο θα έπρεπε να βρίσκεται σχετικά μπροστά στη λίστα, έτσι ώστε να μειωθεί το μέσο κόστος αναζήτησης.

Move-to-front: κάθε φορά που αναζητώ κάποιο στοιχείο α_i το μετακινώ στην 1η θέση της λίστας και μετατοπίζω κάθε στοιχείο της λίστας από την θέση 1 έως $l_i - 1$ κατά μία θέση πιο δεξιά.

Ανάλυση Move-to-front

- Έστω δύο στοιχεία α_i, α_j τα οποία έχουν αναζητηθεί στο παρελθόν.
- Το στοιχείο α_i θα βρίσκεται μπροστά από το α_j εάν το α_i έχει αναζητηθεί μετά την τελευταία αναζήτηση του α_j.
- Αν δεν έχουν αναζητηθεί καθόλου, η τρέχουσα σχετική τους τοποθέτηση είναι σύμφωνα με την αρχική τους τοποθέτηση στην λίστα.
- Η πιθανότητα αυτή, P_{ij}, δηλαδή το α_iνα βρίσκεται πριν το α_j, είναι $\frac{p_i}{(p_i+p_j)}$.
- Η αντίστοιχη πιθανότητα, P_{ji}, να βρίσκεται το α_jπριν το α_iείναι $\frac{p_j}{(p_i+p_j)}$.
- Απόδειξη για την δεύτερη περίπτωση (ομοίως και για την πρώτη):
- Έστω μία ακολουθία αναζητήσεων Search(α_{i_k}) (k=1,…,r) και αναζητούμε την πιθανότητα το στοιχείο α_j να είναι το τελευταίο που αναζητήθηκε σε σχέση με το α_i.
- Υποθέτουμε το r αρκετά μεγάλο. Έστω Search(α_{i_l}) η τελευταία λειτουργία αναζήτησης που αναζήτησε το στοιχείο α_j, δηλ. α_{i_l}= α_j.
- Η πιθανότητα για αυτό το γεγονός: $p_j(1-p_i-p_j)^{r-l}$ αφού στις τελευταίες $r-l$ αναζητήσεις δεν αναζητήθηκαν τα στοιχεία α_i και α_j.

Ανάλυση Move-to-front

- Αν $l = 0$, αυτό σημαίνει ότι τα στοιχεία α_j και α_i ποτέ δεν αναζητήθηκε στην ακολουθία των r λειτουργιών αναζήτησης.
- Το γεγονός συμβαίνει με πιθανότητα $(1 - p_i - p_j)^r$.
- Σε αυτή την περίπτωση, η σχετική θέση των α_j και α_i εξαρτάται από την αρχική θέση των στοιχείων αυτών στη λίστα.
- Έστω P_{init} η πιθανότητα το α_j να είχε τοποθετηθεί αρχικά αριστερά του α_i στην λίστα.

Επομένως:

$$P_{ji} = \sum_{l=1}^{r} p_j(1 - p_i - p_j)^{r-l} + (1 - p_i - p_j)^r P_{init} = p_j \sum_{l=1}^{r} (1 - p_i - p_j)^{r-l} + (1 - p_i - p_j)^r P_{init} =$$

$$p_j \sum_{l=0}^{r-1} (1 - p_i - p_j)^l + (1 - p_i - p_j)^r P_{init} = p_j \frac{(1-p_i-p_j)^r - 1}{(1-p_i-p_j)-1} + (1 - p_i - p_j)^r P_{init} = p_j \frac{1-(1-p_i-p_j)^r}{p_i+p_j} + (1 - p_i - p_j)^r P_{init}$$

- Έχουμε υποθέσει ότι το r είναι αρκετά μεγάλο. Άρα:
- $\lim_{r \to +\infty} P_{ji} = \lim_{r \to +\infty} \left[p_j \frac{1-(1-p_i-p_j)^r}{p_i+p_j} + (1 - p_i - p_j)^r P_{init} \right] = \frac{p_j}{p_i+p_j}$

Ανάλυση Move-to-front

- Έστω X_{ji} η δείκτρια μεταβλητή του γεγονότος το στοιχείο α_j να είναι πιο πριν το α_i στην λίστα Α. Ισχύει $\mathrm{E}[\mathrm{X}_{ji}] = P_{ji}$.
- Επίσης για το την τρέχουσα θέση l_i του στοιχείου α_i ισχύει:

$$l_i = \sum_{j;j\neq i} \mathrm{X}_{ji}+1$$

- Επομένως:

$$\mathrm{E}[l_i] = \mathrm{E}[\sum_{j;j\neq i} \mathrm{X}_{ji}]+1= \sum_{j;j\neq i} \mathrm{E}[\mathrm{X}_{ji}]+1=\sum_{j;j\neq i} P_{ji}+1=\sum_{j;j\neq i} \frac{p_j}{p_i+p_j}+1$$

Ανάλυση Move-to-front

- Άρα το αναμενόμενο κόστος αναζήτησης του αλγορίθμου Move-to-front είναι:
- $\text{Κόστος} = \sum_i p_i \left(1 + \sum_{j;j\neq i} \frac{p_j}{p_i+p_j}\right) = \sum_i p_i + \sum_{i,j;j\neq i} \frac{p_i p_j}{p_i+p_j}$
- Παρατήρηση: για κάθε i, j με $i \neq j$ ο όρος $\frac{p_i p_j}{p_i+p_j}$ εμφανίζεται δύο φορές στο άθροισμα.
- Κάνουμε επίσης την εξής θεώρηση: $p_1 \geq p_2 \geq \cdots \geq p_n$
- Επομένως το κόστος ισούται με:

Ανάλυση Move-to-front

$$\text{Κόστος} = \sum_i p_i + 2 \sum_{j;j<i} \frac{p_i p_j}{p_i + p_j}$$

$$= \sum_i p_i \, (1 + 2 \sum_{j;j<i} \frac{p_j}{p_i + p_j})$$

$$\leq \sum_i p_i \, (1 + 2 \sum_{j;j<i} 1)$$

$$< \sum_i p_i 2i$$

$$= 2 \sum_i p_i i$$

$$= 2 \, Opt$$

- Θεώρημα: Ο αλγόριθμός Move-to-front έχει αναμενόμενο κόστος αναζήτησης το οποίο είναι το πολύ δύο φορές το κόστος της βέλτιστης λύσης (Opt).

Άσκηση: Εύρεση ανάστροφων ζευγών πίνακα

Έστω πίνακας A, μεγέθους n, με $A[i] \in \{1,2,\dots,n\}$ και $i \in \{1,2,\dots,n\}$. Αν ισχύει $i < j$ και $A[i] > A[j]$, τότε το ζεύγος των (i,j) ονομάζεται ανάστροφο. Έστω ότι τα στοιχεία του πίνακα είναι μία ομοιόμορφη μετάθεση των στοιχείων του συνόλου $\{1,2,\dots,n\}$. Πόσα ανάστροφα ζεύγη ενδέχεται να υπάρχουν στον πίνακα A;

Εύρεση ανάστροφων ζευγών πίνακα

- Πίνακας A μεγέθους n, με $A[i] \in \{1,2,\dots,n\}$ και $i \in \{1,2,\dots,n\}$.

- Ανάστροφο ζεύγος (i,j):

$$i < j \text{ και } A[i] > A[j]$$

Παράδειγμα:

x1	x2	x3	x4	x5	x6	x7	x8
2	4	8	1	3	7	5	6

(x3,x4)

(x6,x7)

(x2,x4)

Εύρεση ανάστροφων ζευγών πίνακα

- Ορίζω τυχαία μεταβλητή X_{ij} με:

$X_{ij} = 1$ όταν $i < j$ και $A[i] > A[j]$ αλλιώς 0.

- Ποια είναι η πιθανότητα ένα ζεύγος να είναι ανάστροφο;

$$\Pr\{X_{ij} = 1\} = ?$$

Εύρεση ανάστροφων ζευγών πίνακα

- Ορίζω τυχαία μεταβλητή X_{ij} με:

$X_{ij} = 1$ όταν $i < j$ και $A[i] > A[j]$

- Ποια είναι η πιθανότητα ένα ζεύγος να είναι ανάστροφο;

$$\Pr\{X_{ij} = 1\} = 1/2$$

Συνεπώς $\mathrm{E}[X_{ij}] = 1/2$

Εύρεση ανάστροφων ζευγών πίνακα

- Έστω X η τυχαία μεταβλητή που εκφράζει τον αριθμό των ανάστροφων ζευγών σε μία ακολουθία μεγέθους n:

$$\mathrm{X} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}$$

Αυτό που θέλουμε να υπολογίσουμε είναι η μέση τιμή της X ($\mathrm{E}[\mathrm{X}]$)

Εύρεση ανάστροφων ζευγών πίνακα

$$E[X] = E[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}]$$

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E[X_{ij}]$$

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} 1/2$$

$$= \frac{n(n-1)}{2} \cdot \frac{1}{2}$$

$$= \frac{n(n-1)}{4}$$

- $\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E[X_{ij}]$

  Για i=1 δημιουργούνται n-1 πιθανά ζεύγη $(X_{12}, X_{13}, X_{14}, \ldots, X_{1n})$

  Για i=2 δημιουργούνται n-2 πιθανά ζεύγη $(X_{23}, X_{24}, X_{25}, \ldots, X_{2n})$

  Για i=3 δημιουργούνται n-3 πιθανά ζεύγη $(X_{34}, X_{35}, X_{36}, \ldots, X_{3n})$

  $\vdots$

  Για i=n-1 δημιουργείται 1 πιθανό ζεύγος (X_{n-1n})

- Επομένως αθροιστικά το σύνολο των πιθανών ζευγών είναι:

$$1 + 2 + 3 + \cdots + (n-2) + (n-1) = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}$$

Άσκηση: Τυχαιοκρατική Ταξινόμηση με τη μέθοδο «διάμεσος-των-τριών»

Ένας τρόπος για τη βελτίωση του αλγορίθμου της Τυχαιοκρατικής Ταξινόμησης (Randomized-Quicksort) είναι η διαμέριση γύρω από ένα στοιχείο οδηγό (pivot), το οποίο θα επιλεγεί πιο προσεκτικά σε σχέση με την τυχαία επιλογή ενός στοιχείου του πίνακα. Μια συνήθης πρακτική είναι η μέθοδος που καλείται **διάμεσος-των-τριών** (**median-of-3**): επίλεξε ως οδηγό τη διάμεσο από ένα σύνολο τριών στοιχείων τα οποία έχουν επιλεγεί με τυχαία σειρά. Για το συγκεκριμένο πρόβλημα, υποθέτουμε ότι ο πίνακας $A[1..n]$ αποτελείται από διαφορετικά μεταξύ τους στοιχεία και ότι $n \geq 3$. Έστω $A'[1..n]$ ο ταξινομημένος πίνακας (η έξοδος της Τυχαιοκρατικής Ταξινόμησης). Με εφαρμογή της μεθόδου διάμεσος-των-τριών για την επιλογή του στοιχείου οδηγού x, ορίζουμε την πιθανότητα $p_i = \Pr\{x = A'[i]\}$.

Α) Ορίστε την πιθανότητα p_i ως συνάρτηση του n και i, όπου $i = 2,3, \dots, n-1$. (Σημειώνεται ότι $p_1 = p_n = 0$)

Β) Πόσο έχει αυξηθεί η πιθανότητα να επιλεγεί ως στοιχείο οδηγός το $x = A'\left[\lfloor\frac{(n+1)}{2}\rfloor\right]$ η διάμεσος του προς ταξινόμηση πίνακα $A[1..n]$; Βρείτε το όριο του λόγου των δύο αυτών πιθανοτήτων, υποθέτοντας ότι $n \to \infty$.

Γ) Αν ορίσουμε μια «καλή» διαμέριση ως αυτήν για την οποία επιλέγουμε ως $x = A'[i]$, όπου $\frac{n}{3} \leq i \leq \frac{2n}{3}$, πόσο αυξάνεται η πιθανότητα να λάβουμε μια «καλή» διαμέριση σε σχέση με την κανονική υλοποίηση; (Προσεγγίστε το άθροισμα με ολοκλήρωμα).

Τυχαιοκρατική Ταξινόμηση

Α) $p_i = \frac{(i-1)(n-i)}{\binom{n}{3}} = \frac{(i-1)(n-i)}{\frac{n(n-1)(n-2)}{6}} = 6\frac{(i-1)(n-i)}{n(n-1)(n-2)}$ αφού σίγουρα στην επιλεγμένη τριάδα στοιχειών θα πρέπει να είναι επιλεγμένο το i-οστο μικρότερο στοιχείο του πίνακα Α και για τα άλλα δύο στοιχεία, το ένα πρέπει να είναι μικρότερο και το άλλο μεγαλύτερο από το i-οστό μικρότερο στοιχείο προκειμένου το στοιχείο αυτό να επιλεχθεί ως διάμεσος τελικά.

Β) Για $i = \lfloor\frac{n+1}{2}\rfloor$, έχουμε $p_i = 6\frac{(i-1)(n-i)}{n(n-1)(n-2)} \geq 6\frac{\left(\frac{n}{2}-1\right)\left(n-\frac{n+1}{2}\right)}{n(n-1)(n-2)} = \frac{6\left(\frac{n}{2}-1\right)\left(\frac{n}{2}-\frac{1}{2}\right)}{n(n-1)(n-2)} = \frac{6n^2\left(\frac{1}{2}-\frac{1}{n}\right)\left(\frac{1}{2}-\frac{1}{2n}\right)}{n^3\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)} \Rightarrow \lim_{n\to\infty} p_i = \lim_{n\to\infty} \frac{6\frac{1}{2}\frac{1}{2}}{n} = \lim_{n\to\infty}\left(\frac{6}{4}\frac{1}{n}\right) = \lim_{n\to\infty}\left(\frac{3}{2}\frac{1}{n}\right)$. Άρα $\lim_{n\to\infty} \frac{p_i}{\frac{1}{n}} = \frac{3}{2}$.

Τυχαιοκρατική Ταξινόμηση

Γ)) Η πιθανότητα η μέθοδος διάμεσος-των-τριών να επιλέξει ένα pivot ένα στοιχείο μεταξύ του $\frac{n}{3}$ - μικρότερου και του $\frac{2n}{3}$ - μικρότερου στοιχείου του πίνακα Α είναι:

$$\sum_{i=\frac{n}{3}}^{\frac{2n}{3}} p_i =$$

$$\sum_{i=\frac{n}{3}}^{\frac{2n}{3}} 6\frac{(i-1)(n-i)}{n(n-1)(n-2)} =$$

$$\frac{6}{n(n-1)(n-2)}\sum_{i=\frac{n}{3}}^{\frac{2n}{3}}(i-1)(n-i) =$$

$$\frac{6}{n(n-1)(n-2)}\sum_{i=\frac{n}{3}}^{\frac{2n}{3}}(i\cdot n - i^2 - n + i) =$$

$$\frac{6}{n(n-1)(n-2)}\sum_{i=\frac{n}{3}}^{\frac{2n}{3}}(i(n+1) - i^2 - n)$$

Τυχαιοκρατική Ταξινόμηση

- Αν επιλέγουμε το στοιχείο οδηγό όπως ο βασικός quicksort, η πιθανότητα το pivot να είναι μεταξύ του $\frac{n}{3}$-μικρότερου και του $\frac{2n}{3}$-μικρότερου (συμπεριλαμβανομένων και αυτών των στοιχείων) είναι:

$$\frac{1}{n}\left(\frac{2n}{3}-\frac{n}{3}+1\right)=\frac{1}{n}\left(\frac{n}{3}+1\right)=\left(\frac{1}{3}+\frac{1}{n}\right).$$

Το τελικό ζητούμενο της άσκησης είναι ο υπολογισμός του λόγου

$$\frac{\sum_{i=\frac{n}{3}}^{\frac{2n}{3}} p_i}{\left(\frac{1}{3}+\frac{1}{n}\right)}$$

Άσκηση: Σταθμισμένη Διάμεσος

Έστω n διακριτά στοιχεία $x_1, x_2, \cdots, x_n$ με αντίστοιχα θετικά βάρη $w_1, w_2, \cdots, w_n$ τέτοια ώστε $\sum_{i=1}^{n} w_i = 1$.

Ονομάζεται **σταθμισμένη (μικρότερη) διάμεσος** (weighted lower median) το στοιχείο x_k για το οποίο ισχύει:

$$\sum_{x_i<x_k} w_i < \frac{1}{2} \quad \text{και} \quad \sum_{x_i>x_k} w_i \leq \frac{1}{2}$$

- Για παράδειγμα αν τα στοιχεία είναι 3, 8, 2, 5, 4, 1, 6 και τα βάρη αντίστοιχα είναι 0.12 , 0.35 , 0.025 , 0.08 , 0.15 , 0.075, 0.2 τότε η διάμεσος είναι το 4 αλλά σταθμισμένη διάμεσος είναι το 6.
- 1 2 3 4 5 6 8
- 0.075 0.025 0.12 0.15 0.08 0.2 0.35

Σταθμισμένη Διάμεσος

Α. Ισχύει ότι η διάμεσος των $x_1, x_2, \cdots, x_n$ είναι η σταθμισμένη διάμεσος των x_i με βάρη $w_i = 1/n$ για κάθε $i = 1,2, \cdots, n$;

Δικαιολογείστε την απάντησή σας.

Β. Δείξτε πως μπορώ να βρω τη σταθμισμένη διάμεσο n στοιχείων με χρόνο χειρότερης περίπτωσης $O(nlogn)$, χρησιμοποιώντας την ταξινόμηση.

Γ. Δείξτε πως μπορείτε να υπολογίσετε τη σταθμισμένη διάμεσο με χρόνο χειρότερης περίπτωσης $\Theta(n)$, χρησιμοποιώντας ένα γραμμικό αλγόριθμο εύρεσης της διαμέσου, όπως ο αλγόριθμος SELECT της παραγράφου 9.3 του βιβλίου.

Σταθμισμένη Διάμεσος - Ερώτημα Α

- Έστω m_k ο αριθμός των στοιχείων x_i που είναι μικρότερα από το x_k. Άρα ο x_k είναι στην θέση m_k+1 του ταξινομημένου πίνακα των x_i
- Όταν αναθέτουμε βάρος $1/n$ σε κάθε στοιχείο x_i ισχύει ότι:

$$\sum_{x_i<x_k} w_i = m_k/n$$

Και

$$\sum_{x_i>x_k} w_i = (n - m_k - 1)/n$$

Σταθμισμένη Διάμεσος - Ερώτημα Α

Θα πρέπει όμως:

$$\sum_{x_i<x_k} w_i = \frac{m_k}{n} < 1/2$$

Και

$$\sum_{x_i>x_k} w_i = (n - m_k - 1)/n \leq 1/2$$

Επομένως n/2-1 $\leq m_k <$ n/2.

Η μόνη τιμή του m_k για την οποία ισχύουν και οι 2 ανισώσεις είναι:

$$m_k = \lceil n/2 \rceil - 1$$

Επομένως η θέση του στοιχείου x_k είναι $\lceil n/2 \rceil$. Άρα η σταθμισμένη διάμεσος ταυτίζεται με τη διάμεσο (έχουμε ίδιο αριθμό στοιχείων που είναι μικρότερα και μεγαλύτερα της σταθμισμένης διαμέσου).

Σταθμισμένη Διάμεσος - Ερώτημα Β

- Αρχικά χρησιμοποιούμε τον αλγόριθμο mergesort για να ταξινομήσουμε τα x_i με βάση την τιμή τους σε χρόνο $O(nlogn)$.
- Έστω S_i το άθροισμα των βαρών των πρώτων i στοιχείων των ταξινομημένων στοιχείων. Σημειώνεται ότι χρειάζεται χρόνος O(1) για να ενημερωθεί η τιμή του S_i .
- Υπολογείστε τα $S_1, S_2, \cdots$ μέχρι να φτάσουμε στο k-οστό στοιχείο για το οποίο ισχύει ότι:
  - $S_{k-1} < 1/2$ και $S_k \geq 1/2$
- Τότε η σταθμισμένη διάμεσος είναι το στοιχείο x_k.

Υπενθύμιση: Αλγόριθμος SELECT

Ο αλγόριθμος SELECT εντοπίζει το i-οστο μικρότερο στοιχείο ενός πίνακα με $n > 1$ διακριτά στοιχεία, ακολουθώντας τα εξής βήματα:

1. Διαχωρίζει τα n στοιχεία του πίνακα σε $\lfloor n/5 \rfloor$ υποπίνακες των 5 στοιχείων ο καθένας, και το πολύ ένας από τους υποπίνακες θα περιέχει τα υπόλοιπα $n \; mod \; 5$ στοιχεία.
2. Βρίσκουμε τη διάμεσο κάθε υποπίνακα ($\lceil n/5 \rceil$ το πλήθος). Ταξινομούμε πρώτα τον κάθε υποπίνακα με χρήση του αλγορίθμου *insertionSort* (το πολύ 5 στοιχεία ο κάθε υποπίνακας) και έπειτα παίρνουμε τη διάμεσο του κάθε υποπίνακα.
3. Χρησιμοποιούμε τον αλγόριθμο SELECT αναδρομικά για να βρούμε τη διάμεσο των $\lceil n/5 \rceil$ διαμέσων που εντοπίσαμε στο βήμα 2 (αν υπάρχει άρτιος αριθμός διαμέσων, κατά σύμβαση παίρνουμε τη μικρότερη διάμεσο).
4. Διαχωρίζουμε τον αρχικό πίνακα με βάση τη διάμεσο των διαμέσων x που βρέθηκε στο προηγούμενο βήμα. Έστω k το πλήθος των μικρότερων στοιχείων στοιχείων που προέκυψαν από το διαχωρισμό του πίνακα αυξημένο κατά 1. Έτσι ώστε το στοιχείο x να είναι το k-οστο μικρότερο και να βρίσκονται $n - k$ στοιχεία στον άλλο υποπίνακα.
5. Αν $i = k$ τότε επιστρέφω το x. Διαφορετικά, χρησιμοποιώ την SELECT αναδρομικά για να βρω το i-οστο μικρότερο στοιχείο στον πρώτο υποπίνακα με τα μικρότερα στοιχεία, αν $i < k$, ή αναζητώ πλέον το $(i - k)$-οστο μικρότερο στοιχείο στο δεύτερο υποπίνακα με τα μεγαλύτερα στοιχεία, αν $i > k$.

Σταθμισμένη Διάμεσος - Ερώτημα Γ

- Αρχικά βρίσκω τη διάμεσο των x και διαχωρίζω τον πίνακα με βάση το x. Υπολογίζω τα $\sum_{x_i<x} w_i$ και $\sum_{x_i>x} w_i$ και ελέγχω αν κάποιο από τα 2 είναι μεγαλύτερο από το ½ . Αν όχι, σταματώ. Αν ναι, συνεχίζω στον υποπίνακα που έχει άθροισμα βαρών μεγαλύτερο από ½.
- Χρόνος εκτέλεσης:

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

Άρα $T(n) = \Theta(n)$.

Άσκηση: Κάτω όριο συγχώνευσης δύο ταξινομημένων λιστών

Το πρόβλημα της συγχώνευσης δύο ταξινομημένων λιστών προκύπτει συχνά. Έχουμε δει μια διαδικασία για αυτό ως υπορουτίνα MERGE στην Ενότητα 2.3.1.

Σε αυτό το πρόβλημα, θα αποδείξουμε ένα κατώτερο όριο $2n - 1$ στον χειρότερο αριθμό συγκρίσεων που απαιτούνται για τη συγχώνευση δύο ταξινομημένων λιστών, καθεμία από τις οποίες περιέχει n στοιχεία.

Πρώτα θα δείξουμε ένα χαμηλότερο όριο συγκρίσεων $2n - \mathrm{O}(n)$ χρησιμοποιώντας ένα δέντρο αποφάσεων.

Κάτω όριο στη συγχώνευση ταξινομημένων λιστών

1. Δεδομένων $2n$ αριθμών, υπολογίστε τον αριθμό όλων των πιθανών τρόπων για να τους χωρίσετε σε δύο ταξινομημένες λίστες, η καθεμία με n αριθμούς.
2. Χρησιμοποιώντας ένα δέντρο αποφάσεων και την απάντησή σας στο ερώτημα (a), δείξτε ότι οποιοσδήποτε αλγόριθμος που συγχωνεύει σωστά δύο ταξινομημένες λίστες πρέπει να εκτελεί τουλάχιστον $2n - o(n)$ συγκρίσεις.
3. Δείξτε ότι εάν δύο στοιχεία είναι διαδοχικά στην συνολική ταξινομημένη σειρά και είναι από διαφορετικές λίστες, τότε πρέπει να συγκριθούν.
4. Χρησιμοποιήστε την απάντησή σας στο προηγούμενο ερώτημα για να δείξετε ένα χαμηλότερο όριο συγκρίσεων $2n - 1$ για τη συγχώνευση δύο ταξινομημένων λιστών.

Κάτω όριο στη συγχώνευση ταξινομημένων λιστών

1. Υπάρχουν $\binom{\boldsymbol{2n}}{\boldsymbol{n}}$ τρόποι να διαχωριστούν $2n$ αριθμοί, σε 2 ταξινομημένες λίστες πλήθους n.

Κάτω όριο στη συγχώνευση ταξινομημένων λιστών

2. Οποιοδήποτε δέντρο απόφασης για τη συγχώνευση 2 ταξινομημένων λιστών μήκους n, θα πρέπει να έχει τουλάχιστον $\binom{\boldsymbol{2n}}{\boldsymbol{n}}$ κόμβους φύλλα, επομένως θα πρέπει να έχει **ύψος τουλάχιστον** $\boldsymbol{\log\binom{2n}{n}}$.

Με βάση τον τύπο του Stirling, δηλαδή $n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n\left(1+\Theta\left(\frac{1}{n}\right)\right)$ ισχύει:

$$\binom{2n}{n} = \frac{2^{2n}}{\sqrt{\pi n}}\left(1+O\left(\frac{1}{n}\right)\right) \Rightarrow$$

$$\lg\binom{2n}{n} = \lg\left(\frac{2^{2n}}{\sqrt{\pi n}}\left(1+O\left(\frac{1}{n}\right)\right)\right)$$

$$= \lg\left(\frac{2^{2n}}{\sqrt{\pi n}}\right) + \lg\left(1+O\left(\frac{1}{n}\right)\right)$$

$$= \lg(2^{2n}) - \lg(\sqrt{\pi n}) + \lg\left(1+O\left(\frac{1}{n}\right)\right)$$

$$= 2n - o(n)$$

Κάτω όριο στη συγχώνευση ταξινομημένων λιστών

3.

- Έστω οι ταξινομημένες λίστες Α και Β n και m στοιχείων αντίστοιχα, και έστω α_i και b_j τα στοιχεία που δεν συγκρίνονται και είναι διαδοχικά στον τελικό πίνακα. Ας θεωρήσουμε ότι $\alpha_i < b_j$
- Γνωρίζουμε ότι $\max\{b_{j-1}, \alpha_{i-1}\} < \alpha_i < b_j < \min\{b_{j+1}, \alpha_{i+1}\}$ αφού τα α_i και b_j πρέπει να είναι διαδοχικά στο τελικό αποτέλεσμα.
- Για να είναι ορθός ο αλγόριθμος, θα πρέπει να τοποθετήσει το στοιχείο από τον πίνακα Α στην (i+j-1)-οστή θέση του τελικού πίνακα και το στοιχείο από το πίνακα Β στη θέση i+j, χωρίς να κάνει σύγκριση των δύο αυτών στοιχείων.
- Παρατηρείστε επίσης, ότι όλες υπόλοιπες συγκρίσεις που κάνει ο αλγόριθμος δεν μπορούν να δώσουν έμμεση πληροφορία για τη σχετική διάταξη των α_i και b_j.
- Αν ανταλλάξουμε τα δύο στοιχεία στις λίστες Α και Β τότε οι λίστες αυτές εξακολουθούν να είναι ταξινομημένες.
- Αφού ο αλγόριθμος δεν συγκρίνει τα α_i και b_j και πάλι ο αλγόριθμος πρέπει να βάλει με την παραπάνω σειρά τα δύο αυτά νέα στοιχεία των λιστών στον τελικό πίνακα. Στην περίπτωση αυτή, το αποτέλεσμα είναι λανθασμένο αφού το b_j θα προηγούταν του α_i στον τελικό πίνακα.
- Άρα τα α_i και b_j πρέπει να συγκριθούν.

Κάτω όριο στη συγχώνευση ταξινομημένων λιστών

4. Έστω οι λίστες

$$A = 1, 3, 5, \ldots, 2n - 1$$

και

$$B = 2, 4, 6, \ldots, 2n$$

Από το ερώτημα c, πρέπει να συγκρίνετε 1 με 2, 2 με 3, 3 με 4 και ούτω καθεξής μέχρι να συγκρίνουμε $2n - 1$ με $2n$.

Αυτό αντιστοιχεί συνολικά σε $2n - 1$ συγκρίσεις.