

Περιεχόμενα

- Αναζήτηση
- Κατακερματισμός
- Ταξινόμηση

Αναζήτηση

❖ Ορισμός

Αναζήτηση ονομάζεται η διαδικασία εύρεσης μίας εγγραφής που ανήκει σε ένα σύνολο δεδομένων, με βάση κάποιο πεδίο της εγγραφής το οποίο ονομάζεται πρωτεύον κλειδί

❖ Γνωστές Μέθοδοι Αναζήτησης

- Σειριακή Αναζήτηση (sequential, linear search)
- Δυαδική Αναζήτηση (binary search)
- Αναζήτηση Fibonacci (Fibonacci search)
- Αναζήτηση Παρεμβολής (interpolation search)
- Αναζήτηση Άλματος (jump search)

Αναζήτηση

❖ Σειριακή Αναζήτηση (sequential, linear search)

Πρόκειται για την πιο απλή μέθοδο αναζήτησης κατά την οποία κάθε στοιχείο της δομής ελέγχεται με το προς αναζήτηση στοιχείο.

Πρόταση 1

Ο αριθμός των συγκρίσεων για την ανεπιτυχή αναζήτηση είναι $A=n$.

Πρόταση 2

Αν οι πιθανότητες αναζήτησης των στοιχείων του πίνακα είναι ίδιες, τότε η μέση τιμή του αριθμού των συγκρίσεων για την επιτυχή αναζήτηση ενός κλειδιού είναι:

$$E = (n+1) / 2;$$

Αναζήτηση

❖ Η σειριακή μέθοδος αναζήτησης είναι η πιο απλή και η λιγότερο αποτελεσματική.

❖ Χρησιμοποιείται κυρίως όταν :

1. Ο πίνακας είναι μη ταξινομημένος.
2. Ο πίνακας είναι μικρού μεγέθους.
3. Η αναζήτηση σε ένα συγκεκριμένο πίνακα γίνεται σπάνια.

Αναζήτηση

❖ Δυαδική Αναζήτηση(binary search)

Εφαρμόζεται σε ταξινομημένους κατά αύξουσα ή φθίνουσα τάξη πίνακες.

Συγκρίνεται η τιμή του ζητούμενου κλειδιού με το κλειδί που βρίσκεται στη μεσαία θέση του πίνακα. Αν είναι ίσα τότε η αναζήτηση είναι επιτυχής. Αν όχι, τότε ο μισός πίνακας εξαιρείται από την αναζήτηση, γιατί αποκλείεται να περιέχει το ζητούμενο κλειδί. Η διαδικασία συνεχίζεται με τον ίδιο τρόπο μέχρι να βρεθεί το κλειδί ή να διαπιστωθεί ότι δεν υπάρχει.

Αναζήτηση

Πρόταση 3

Η μέση τιμή του αριθμού των συγκρίσεων για την επιτυχή ή μη επιτυχή αναζήτηση ενός κλειδιού είναι:

$$E = k - \frac{2^k - k - 1}{n}, \quad k = \lfloor \log n \rfloor$$

$$A = \lceil \log n \rceil$$

Αναζήτηση

❖ Αναζήτηση Fibonacci (Fibonacci search)

Η μέθοδος αυτή χωρίζει τον πίνακα σύμφωνα με την ακολουθία αριθμών Fibonacci.

Η μέθοδος αυτή αναζήτησης είναι κατά 4% αργότερη από την δυαδική αναζήτηση, όμως το γεγονός ότι δεν περιέχει την πράξη της διαίρεσης, είναι ένα πολύ καλό πλεονέκτημα της.

Αναζήτηση

❖ Αναζήτηση Παρεμβολής (interpolation search)

Πρόκειται για μέθοδο αναζήτησης η οποία εφαρμόζει μαθηματική παρεμβολή για την εύρεση του προς αναζήτηση κλειδιού.

Αν τα κλειδιά υπακούουν σε ομοιόμορφη κατανομή τότε, η μέθοδος :

- Έχει την καλύτερη απόδοση μεταξύ όλων των μεθόδων αναζήτησης .
- Είναι πολύ σταθερή.
- Η μέση τιμή των απαραίτητων συγκρίσεων για την επιτυχή ή ανεπιτυχή αναζήτηση είναι τάξης $O(\log \log n)$.

Αναζήτηση

Μειονεκτήματα

1. Αν η κατανομή των κλειδιών απέχει από την ομοιόμορφη, τότε η επίδοσή της μεθόδου εκφυλίζεται σε τάξης $O(n)$.
2. Ο τρόπος υπολογισμού της επόμενης θέσης όπου θα γίνει η σύγκριση είναι ιδιαίτερα χρονοβόρος.
3. Αν η κατανομή των κλειδιών δεν είναι γνωστή τότε η μέθοδος αυτή δεν είναι σταθερή και επομένως είναι αναξιόπιστη.
4. Δεν μπορεί να χρησιμοποιηθεί σε πίνακα με επαναλαμβανόμενα κλειδιά

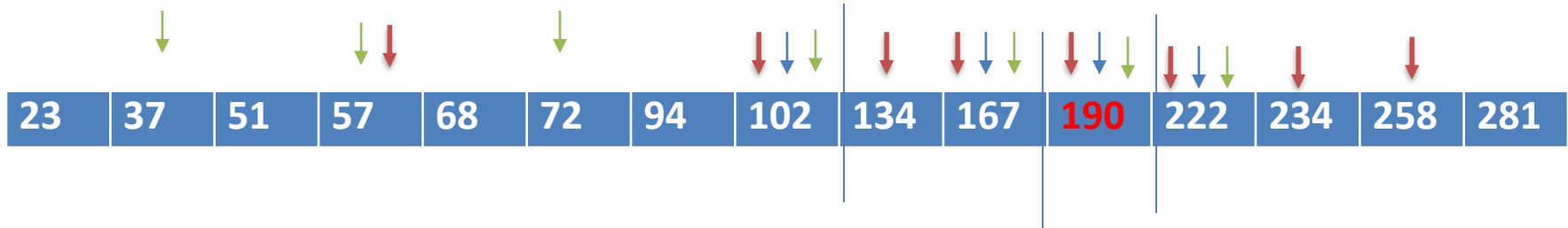
Αναζήτηση

❖ Αναζήτηση Άλματος (jump search)

Η αναζήτηση αρχίζει από την αρχή της δομής και συνεχίζεται με συγκρίσεις κλειδιών που απέχουν μία καθορισμένη απόσταση, το λεγόμενο **άλμα**, μεγέθους n_1 . Όταν βρεθεί το κλειδί με τιμή μεγαλύτερη (δεδομένου ότι τα κλειδιά είναι ταξινομημένα κατά αύξουσα σειρά) από το ζητούμενο κλειδί, τότε η αναζήτηση συνεχίζει προς τα πίσω με βήμα n_2 , μέχρι την προσπέλαση του ζητούμενου κλειδιού.

Αναζήτηση

Άσκηση 1: Ποιος ο αριθμός των συγκρίσεων που απαιτείται για την επιτυχή αναζήτηση του κλειδιού 190 στην παρακάτω δομή, με την εφαρμογή της σειριακής αναζήτησης, της δυαδικής αναζήτησης και της αναζήτησης άλματος; Για την τρίτη μέθοδο θεωρείστε 2 περιπτώσεις αναζήτησης, για $n_1=2$, $n_2=1$ και $n_1=4$, $n_2=3$.



↓ Δυαδική Αναζήτηση

↓ Αναζήτηση άλματος: $n_1=2$, $n_2=1$

↓ Αναζήτηση άλματος: $n_1=4$, $n_2=3$

Κατακερματισμός

❖ **Κατακερματισμός (hashing)** ονομάζεται η διαδικασία μετασχηματισμού ενός κλειδιού σε θέση μνήμης (key-to-address transformation).

❖ Επιτυγχάνεται με τη βοήθεια της **συνάρτησης κατακερματισμού (hashing function)**, που έχει ως σκοπό την αντιστοίχιση του πεδίου τιμών των κλειδιών στο πεδίο των τιμών των διευθύνσεων κατά τυχαίο τρόπο.

❖ Καλή συνάρτηση είναι εκείνη που διασπείρει τις εγγραφές σε όλη την έκταση του πίνακα, ακόμη και αν οι τιμές των κλειδιών συγκεντρώνονται σε μερικές περιοχές του διαστήματος των κλειδιών.

❖ Οι αλγόριθμοι αναζήτησης που χρησιμοποιούν κατακερματισμό αποτελούνται από δύο χωριστά μέρη:

1. Τον υπολογισμό μίας συνάρτησης κατακερματισμού και
2. Την διαδικασία **επίλυσης συγκρούσεων (collision resolution)**, στην οποία επιλύεται το πρόβλημα της αντιστοίχισης δύο διαφορετικών κλειδιών στην ίδια θέση μνήμης.

Κατακερματισμός

❖ Διαχείριση Συγκρούσεων

Αν μία συνάρτηση κατακερματισμού διακρίνεται από μεγάλο αριθμό συγκρούσεων, τότε λέγεται ότι παρουσιάζει το φαινόμενο της **συγκέντρωσης** (clustering).

Οι τεχνικές για τη λύση του προβλήματος των συγκρουόμενων κλειδιών διακρίνονται στις εξής τρεις κατηγορίες:

1. Με Ανοικτή Διεύθυνση
2. Με Αλυσίδες
3. Με Ψευδοαλυσίδες

Κατακερματισμός

Μέθοδοι Ανοικτής Διεύθυνσης

1. **Γραμμική αναζήτηση (linear search)** – Οποιαδήποτε και να είναι η αρχικά χρησιμοποιούμενη συνάρτηση κατακερματισμού, $\text{hash}(\text{key})$, η μέθοδος της γραμμικής αναζήτησης αποθηκεύει το κλειδί που συγκρούεται στην πρώτη επόμενη διαθέσιμη θέση.
2. **Τετραγωνική αναζήτηση (quadratic search)** – Οι επόμενες θέσεις που θα εξετασθούν μπορούν να υπολογιστούν με βάση τη σχέση:

$$\left(c_1 \times i^2 + c_2 \times i + \text{key} \text{ MOD } m \right) \text{MOD } m$$

όπου i είναι ο αύξων αριθμός αναζήτησης ($i=0,1,2,\dots$), ενώ τα c_1, c_2 είναι αυθαίρετες σταθερές.

Κατακερματισμός

❖ Τέλειος Κατακερματισμός

Μία συνάρτηση κατακερματισμού λέγεται **τέλεια (perfect)** όταν για κάθε ζεύγος κλειδιών $key1 \leftrightarrow key2$ ισχύει $hash(key1) \leftrightarrow hash(key2)$.

Η εύρεση μία τέλειας συνάρτησης κατακερματισμού δεν είναι εύκολη διαδικασία. Επιπλέον μία συνάρτηση είναι τέλεια για ένα σύνολο κλειδιών, το οποίο αν αλλάξει η συνάρτηση χάνει την ιδιότητάς της.

Κατακερματισμός

Άσκηση 2: Χρησιμοποιώντας την συνάρτηση κατακερματισμού “ $\text{hash}(\text{key}) = \text{key} \text{ MOD } n$ ”, αποθηκεύστε τα κλειδιά 34, 7, 53, 21, 49, 77, 87, 2 με τη σειρά που δίνονται, σε πίνακα 9 θέσεων, χρησιμοποιώντας τη μέθοδο της ανοικτής διεύθυνσης (ή υπερχείλισης) ως τεχνική διαχείρισης συγκρούσεων.

(34): $34 \bmod 9 = 7$

(7): $7 \bmod 9 = 7$, σύγκρουση $\rightarrow 8$

(53): $53 \bmod 9 = 8$, σύγκρουση $\rightarrow 0$

(21): $21 \bmod 9 = 3$

(49): $49 \bmod 9 = 4$

(77): $77 \bmod 9 = 5$

(87): $87 \bmod 9 = 6$

(2): $2 \bmod 9 = 2$

0	1	2	3	4	5	6	7	8
53		2	21	49	77	87	34	7

Κατακερματισμός

Άσκηση 2: Χρησιμοποιώντας την συνάρτηση κατακερματισμού “hash(key)=key MOD n”, αποθηκεύστε τα κλειδιά 34, 7, 53, 21, 49, 77, 87, 2 με τη σειρά που δίνονται, σε πίνακα 9 θέσεων, χρησιμοποιώντας τη μέθοδο της τετραγωνικής αναζήτησης ($C_1=1$, $C_2=2$) ως τεχνική διαχείρισης συγκρούσεων.

$$(34)-(i=0): (34 \bmod 9) \bmod 9 = 7 \bmod 9$$

$$(7)-(i=0): (7 \bmod 9) \bmod 9 = 7, \text{σύγκρουση}$$

$$(7)-(i=1): (1+2+7 \bmod 9) \bmod 9 = 10 \bmod 9 = 1$$

$$(53)-(i=0): (53 \bmod 9) \bmod 9 = 8 \bmod 9 = 8$$

$$(21)-(i=0): (21 \bmod 9) \bmod 9 = 3 \bmod 9 = 3$$

$$(49)-(i=0): (49 \bmod 9) \bmod 9 = 4 \bmod 9 = 4$$

$$(77)-(i=0): (77 \bmod 9) \bmod 9 = 5 \bmod 9 = 5$$

$$(87)-(i=0): (87 \bmod 9) \bmod 9 = 6 \bmod 9 = 6$$

$$(2)-(i=0): (2 \bmod 9) \bmod 9 = 2 \bmod 9 = 2$$

0	1	2	3	4	5	6	7	8
	7	2	21	49	77	87	34	53

$$(c_1 \times i^2 + c_2 \times i + \text{key} \bmod m) \bmod m$$

Ταξινόμηση

❖ **Ταξινόμηση (sorting)** ονομάζεται η διαδικασία τοποθέτησης ενός συνόλου εγγραφών σε μία ιδιαίτερη σειρά (συνήθως **αύξουσα τάξη – ascending sequence**) με βάση το μέγεθος του πρωτεύοντος κλειδιού εγγραφής.

❖ Σκοπός της ταξινόμησης είναι η διευκόλυνση της αναζήτησης των εγγραφών του ταξινομηθέντος συνόλου.

❖ Ορισμός

Δοθέντων των στοιχείων a_1, a_2, \dots, a_n , η ταξινόμηση συνίσταται στη μετάθεση (permutation) της θέσης των στοιχείων, ώστε να τοποθετηθούν σε μία σειρά $a_{1_k}, a_{2_k}, \dots, a_{n_k}$, έτσι ώστε δοθείσης μίας **συνάρτησης διάταξης (ordering function), f** , να ισχύει

$$f(a_{k_1}) \leq f(a_{k_2}) \leq \dots \leq f(a_{k_n})$$

Ταξινόμηση

- ❖ Με την κατάλληλη επιλογή της συνάρτησης f , μπορούμε να πετύχουμε **φθίνουσα τάξη (descending sequence)** μεγέθους του κλειδιού, ενώ στην γενική περίπτωση η ταξινόμηση μπορεί να στηριχτεί σε δύο ή περισσότερα κλειδιά εγγραφής.
- ❖ Οι μέθοδοι ταξινόμησης χωρίζονται στις εσωτερικές (internal – ταξινόμηση δεδομένων που βρίσκονται στην γρήγορη και άμεσης πρόσβασης κύρια μνήμη) και τις εξωτερικές (external – ταξινόμηση δεδομένων που είναι αποθηκευμένα σε περιφερειακές συσκευές).
- ❖ Μία μέθοδος λέγεται **ευσταθής (stable)**, αν η σχετική διάταξη των στοιχείων με ίσα κλειδιά παραμένει αμετάβλητη με την εφαρμογή της μεθόδου.

Ταξινόμηση

- ❖ Το πιο βασικό κριτήριο της επίδοσης μίας μεθόδου είναι ο αριθμός **C** που μετρά τις απαιτούμενες συγκρίσεις κλειδιών (key comparisons), που εκτελούνται μέχρι να τελειώσει η ταξινόμηση.
- ❖ Ένα άλλο κριτήριο είναι ο αριθμός **M** που μετρά τις μετακινήσεις (moves) των στοιχείων.
- ❖ Οι αριθμοί **C**, **M** είναι συναρτήσεις του αριθμού n των στοιχείων που πρέπει να ταξινομηθούν.

Ταξινόμηση

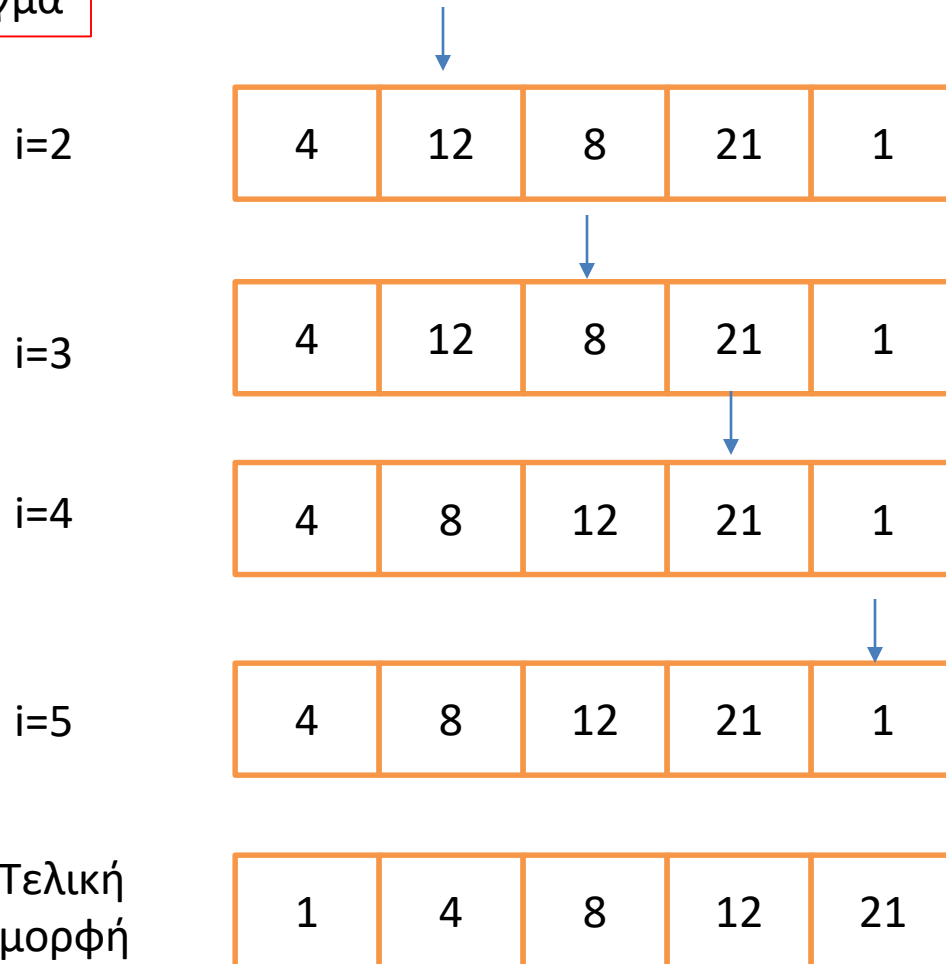
- ❖ Γνωστοί Αλγόριθμοι Ταξινόμησης:
 - Ταξινόμηση ευθείας **εισαγωγής**
 - Ταξινόμηση ευθείας **ανταλλαγής ή μετάθεσης**
 - Ταξινόμηση ευθείας **επιλογής**

❖ Ταξινόμηση Ευθείας Εισαγωγής

Σε κάθε βήμα αρχίζοντας με $i=2$ και αυξάνοντας διαδοχικά το i κατά ένα, το στοιχείο με δείκτη i λαμβάνεται και μεταφέρεται στην κατάλληλη θέση (διαδοχικές συγκρίσεις και μετακινήσεις) της ακολουθίας προορισμού.

Ταξινόμηση

Παράδειγμα



Ταξινόμηση

❖ Πρόταση 4

Οι ελάχιστες, οι μέσες και οι μέγιστες τιμές του αριθμού των συγκρίσεων και των μετακινήσεων κατά την ταξινόμηση ευθείας εισαγωγής είναι:

$$\begin{array}{lll} C_{\min} = n - 1 & C_{ave} = \frac{n^2 + n - 2}{4} & C_{\max} = \frac{n^2 + n}{2} - 1 \\ M_{\min} = 2(n - 1) & M_{ave} = \frac{n^2 + 9n - 10}{4} & M_{\max} = \frac{n^2 + 3n - 4}{2} \end{array}$$

οι ελάχιστες όταν τα στοιχεία είναι ήδη ταξινομημένα και οι μέγιστες όταν τα στοιχεία είναι σε αντίστροφη σειρά.

Ταξινόμηση

❖ Ταξινόμηση Ευθείας Ανταλλαγής

Σύμφωνα με τη μέθοδο αυτή κάθε φορά γίνονται διαδοχικές προσπελάσεις στον πίνακα και μετακινείται το μικρότερο κλειδί της ακολουθίας προς το αριστερό άκρο.

❖ Η μέθοδος αυτή ονομάζεται **ταξινόμηση φυσσαλίδας (bubblesort)**, γιατί αν θεωρήσουμε τον πίνακα σε κατακόρυφη θέση (δεξαμενή με νερό), τότε κάθε προσπέλαση έχει ως αποτέλεσμα την άνοδο της φυσσαλίδας (μικρότερο κλειδί) προς τα επάνω .

Ταξινόμηση

Παράδειγμα

	i=2	i=3	i=4	Τελική Μορφή
4	1	1	1	1
12	4	4	4	4
8	12	8	8	8
21	8	12	12	12
1	21	21	21	21

Ταξινόμηση

❖ Πρόταση 5

Οι ελάχιστες, οι μέσες και οι μέγιστες τιμές του αριθμού των συγκρίσεων και των μετακινήσεων κατά την ταξινόμηση ευθείας ανταλλαγής είναι:

$$C = \frac{n^2 - n}{2}$$

$$M_{\min} = 0 \quad M_{ave} = \frac{3(n^2 - n)}{4} \quad M_{\max} = \frac{3(n^2 - n)}{2}$$

οι ίδιες ελάχιστες, μέγιστες και μέσες τιμές προκύπτει από το γεγονός ότι ο αριθμός των συγκρίσεων είναι ανεξάρτητος της αρχικής σειράς των κλειδιών.

Ταξινόμηση

❖ Ταξινόμηση Ευθείας Επιλογής

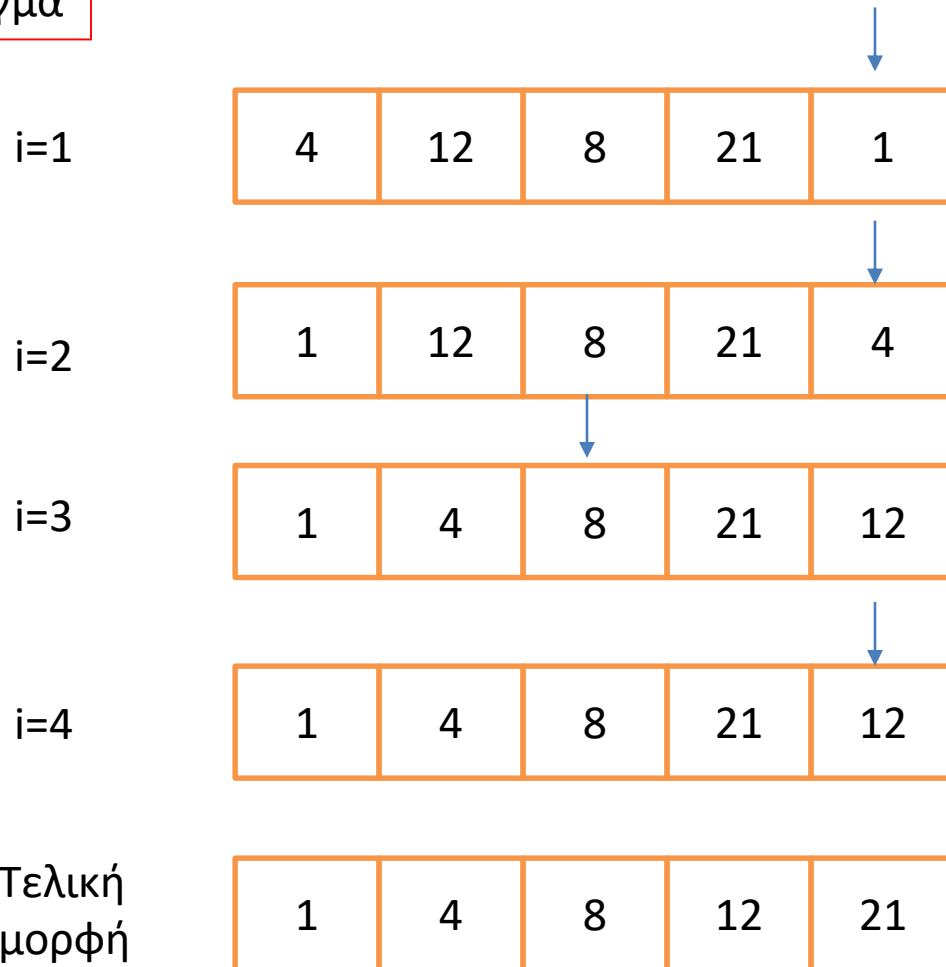
Η μέθοδος αυτή βασίζεται στις παρακάτω δύο αρχές :

1. Επιλογή του στοιχείου με το ελάχιστο κλειδί του μη ταξινομημένου πίνακα (μέρος του πίνακα).
2. Ανταλλαγή αυτού του στοιχείου με το πρώτο στοιχείο του πίνακα (μέρους του πίνακα).

Αυτές οι λειτουργίες επαναλαμβάνονται για τα υπόλοιπα $n-1$ στοιχεία, μέχρι το τέλος να απομείνει μόνο το μεγαλύτερο στοιχείο.

Ταξινόμηση

Παράδειγμα



Ταξινόμηση

❖ Πρόταση 6

Οι ελάχιστες, οι μέσες και οι μέγιστες τιμές του αριθμού των συγκρίσεων και των μετακινήσεων κατά την ταξινόμηση ευθείας επιλογής είναι:

$$C = \frac{n^2 - n}{2}$$

$$M_{\min} = 3(n-1) \quad M_{ave} = n(\ln n + c) \quad c : \text{constant} \quad M_{\max} = \frac{n^2}{4} + 3(n-1)$$

Ταξινόμηση

Άσκηση 3: Ποια μέθοδο ταξινόμησης θα επιλέγατε για την ταξινόμηση 20 κλειδιών αν γνωρίζατε, ότι ο υπολογιστής σας εκτελεί τις πράξεις της σύγκρισης και της μετακίνησης στον ίδιο χρόνο;