3.1 Δεδομένα

Κάθε στοιχείο που γίνεται αντιληπτό με μία από τις πέντε αισθήσεις μας είναι ένα δεδομένο. Τα δεδομένα μπορούν να αναπαραστήσουν αφαιρετικά την πραγματικότητα δηλαδή να μας δείχνουν μία απλοποιημένη όψη της. Π.χ. τα δεδομένα σε ένα αρχείο μαθητών είναι το όνομα, ηλικία, φύλο, τάξη, τμήμα, δεν θα είναι το βάρος, ύψος ή ότι άλλο δεν χρειάζεται να γνωρίζουμε για το μαθητή.
Η επεξεργασία των ακατέργαστων δεδομένων (με κάποιο αλγόριθμο) και ο συσχετισμός τους μας δίνουν τις πληροφορίες βάση των οποίων γίνονται κάποιες ενέργειες οι οποίες μπορούν να παράγουν νέα δεδομένα νέες πληροφορίες, κ.ο.κ.

Η πληροφορική εξετάζει τα δεδομένα από τις ακόλουθες σκοπιές:

- Υλικού
- Γλωσσών προγραμματισμού
- Δομών Δεδομένων
- Ανάλυσης δεδομένων

Μία θέση της μνήμης έχει περιεχόμενο 11110001 .Η τιμή αυτή μπορεί να παριστάνει:

- Τον χαρακτήρα **_** σε κώδικα ASCII.
- Τον χαρακτήρα **ρ** στον κώδικα ΕΛΟΤ 928.
- Την τιμή 241 στο δυαδικό σύστημα.
- Την τιμή -14 στο δυαδικό σύστημα (συμπλήρωμα ως προς 1).
- Την τιμή -15 στο δυαδικό σύστημα (συμπλήρωμα ως προς 2).
- Επίσης αν είναι χαρακτήρας μπορεί να είναι μέρος ενός αλφαριθμητικού.
- Αν είναι αριθμητική τιμή μπορεί να αναπαριστά δεδομένο, διεύθυνση μνήμης ή τον κώδικα μίας εντολής.

3.2 Αλγόριθμοι + Δομές δεδομένων = Προγράμματα

Δομή δεδομένων είναι ένα σύνολο αποθηκευμένων δεδομένων που υφίστανται επεξεργασία από ένα σύνολο λειτουργιών.

Οι βασικές λειτουργίες στους κόμβους των δομών δεδομένων

- ✓ Προσπέλαση: Η πρόσβαση σε ένα κόμβο με σκοπό να τον εξετάσουμε ή να τον τροποποιήσουμε.
- ✓ Εισαγωγή: Η πρόσθεση ενός νέου κόμβου.
- ✓ Διαγραφή: Η αφαίρεση ενός υπάρχοντος κόμβου.
- ✓ Αναζήτηση: Η προσπέλαση των κόμβων με σκοπό τον εντοπισμό κάποιων από αυτούς βάση κάποιας ιδιότητας που έχουν.
- ✓ Ταξινόμηση: Η αναδιάταξη των κόμβων κατά αύξουσα ή φθίνουσα σειρά.
- ✓ Αντιγραφή κόμβων σε άλλη δομή δεδομένων.
- ✓ Συγχώνευση δύο ή περισσότερων δομών δεδομένων σε μία.
- ✓ Διαχωρισμός: Το αντίθετο της συγχώνευσης.

Να γραφεί αλγόριθμος που να δέχεται ως είσοδο το όνομα ενός συνδρομητή του ΟΤΕ και να δίνει ως έξοδο το τηλέφωνό του.

Λύση 1η

Δημιουργείται μία δομή δεδομένων όπου ο κάθε κόμβος της περιέχει όνομα συνδρομητή και τηλέφωνο (O_i, T_i). Οι κόμβοι της δομής δεδομένων είναι συνεχόμενοι (O_1,T_1), (O_2,T_2)…(O_n,T_n). Διαβάζοντας ο αλγόριθμος ένα όνομα κάνει αναζήτηση ώστε να το εντοπίσει σε κάποιο κόμβο το συγκεκριμένο όνομα και να εμφανίσει απ' αυτόν το συγκεκριμένο τηλέφωνο.

Λύση 2η

Η ίδια μορφή δομής δεδομένων μόνο που οι κόμβοι είναι ταξινομημένοι βάση του ονόματος κατά αύξουσα σειρά. Επίσης δημιουργείται και μία δεύτερη δομή δεδομένων όπου κάθε κόμβος της περιέχει το γράμμα του αλφαβήτου και τη θέση του ονόματος που αρχίζει μ' αυτό το γράμμα στη πρώτη δομή δεδομένων, και εδώ οι κόμβοι είναι ταξινομημένοι κατά αύξουσα σειρά.

(A_1,1_1), (B_2,4_2), ($Γ_3$, 9_3)…….($Ω_{24}$, 100_{24})

(Ανδριανού$_1$, 210434..$_1$)……(Βασιλείου$_4$, 32323..$_4$)….
….(Γρηγοριάδη$_9$, 32323..$_9$)……(Ωνάση$_{100}$, 210424..$_{100}$)

Η ***στατική δομή δεδομένων*** έχει σταθερό αριθμό κόμβων που καταλαμβάνουν συνεχόμενες θέσεις στη μνήμη ενός υπολογιστή. Το μέγεθός της ορίζεται στην αρχή του προγράμματος (κατά τη μετάφραση)

Οι **δυναμικές δομές δεδομένων** όπου οι κόμβοι τους δεν καταλαμβάνουν (απαραίτητα) συνεχόμενες θέσεις μνήμης στον υπολογιστή και μπορούν να αλλάξουν το πλήθος των κόμβων τους (αλλάζοντας και το μέγεθος της δομής). Το μέγεθός της δεν ορίζεται στην αρχή του προγράμματος αλλά εκτελείται *δυναμική παραχώρηση μνήμης κατά τη λειτουργία του προγράμματος*

3.3 Πίνακες

Πίνακας είναι μία δομή δεδομένων συνήθως στατική που περιέχει δεδομένα του ίδιου τύπου. Ανάλογα με τη διάσταση συνήθως χρησιμοποιούμε:

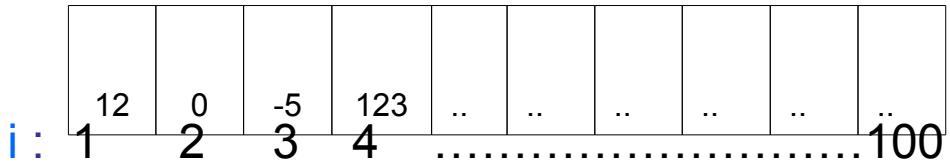
μονοδιάστατους

Σ_1	Σ_2	Σ_3	Σ_4	Σ_5

δισδιάστατους

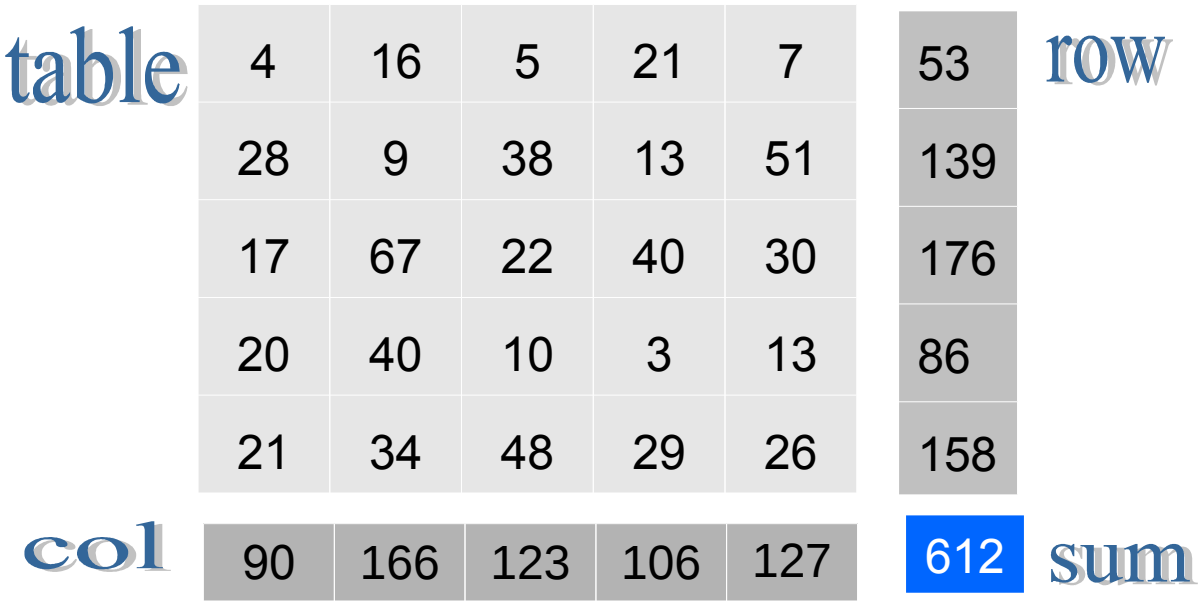
Σ_{11}	Σ_{12}	Σ_{13}	Σ_{14}	Σ_{15}
Σ_{21}	Σ_{22}	Σ_{23}	Σ_{24}	Σ_{25}
Σ_{31}	Σ_{32}	Σ_{33}	Σ_{34}	Σ_{35}

Δίνεται μονοδιάστατος πίνακας table 100 θέσεων που περιέχουν αριθμούς. Να σχεδιαστεί αλγόριθμος που να βρίσκει το ελάχιστο στοιχείο του.




```
Αλγόριθμος Ελαχ_Πίνακα
Δεδομένα // table //
min ← table[ 1 ]
Για i από 2 μέχρι 100
   Αν table[ i ] < min τότε min ← table[ i ]
Τέλος_επανάληψης
Αποτελέσματα // min //
Τέλος Ελαχ_Πίνακα
```

Δίνεται ο δισδιάστατος πίνακας table με m γραμμές και n στήλες. Να βρεθεί το άθροισμα κατά γραμμή, κατά στήλη και συνολικά.




```
Αλγόριθμος Αθρ_Πίνακα
Δεδομένα // m, n, table//
sum←0 ! Αθροιστής για όλα τα στοιχεία του πίνακα table
Για i από 1 μέχρι m
    row[ i ] ← 0 ! Δίνει αρχική τιμή μηδέν στον πίνακα row που
Τέλος_επανάληψης ! θα περιέχει το άθροισμα των γραμμών
Για  j από 1 μέχρι n
    col [ j ] ← 0 ! Δίνει αρχική τιμή μηδέν στον πίνακα col που
Τέλος_επανάληψης ! θα περιέχει το άθροισμα των στηλών
Για i από 1 μέχρι n ! Το i δείκτης για τις στήλες
   Για j από 1 μέχρι m ! Το j δείκτης για τις γραμμές
       sum ←sum + table[ j, i ]
       row[ j ] ← row[ j ] + table[ j, i ]
       col [ i ] ← col[ i ] + table[ j, i]
   Τέλος_επανάληψης
Τέλος_επανάληψης
Αποτελέσματα // row, col, sum//
Τέλος Αθρ_Πίνακα
```

Σε μία εταιρεία εργάζονται 200 υπάλληλοι και είναι γνωστός ο μισθός του καθενός. Να χρησιμοποιηθεί η δομή του πίνακα για να αποθηκεύονται οι μισθοί των υπαλλήλων και να βρεθεί ο κατάλληλος αλγόριθμος υπολογισμού του μεγαλύτερου μισθού.

```
Αλγόριθμος Μεγαλύτερος_μισθός
Διάβασε μισθός[1]
MAX ← μισθός[1]
Για i από 2 μέχρι 200
   Διάβασε μισθός[ i ]
   Αν MAX < μισθός[i] Τοτε   MAX ←μισθός[ i ]
Τέλος_επανάληψης
Αποτελέσματα // MAX //
Τέλος Μεγαλύτερος_μισθός
```

Σε ένα Λύκειο υπάρχουν τρία τμήματα για την Γ' Λυκείου και κάθε τμήμα έχει 35 μαθητές. Να γραφεί αλγόριθμος που θα διαβάζει το μέσο όρο βαθμολογίας κάθε μαθητή και θα υπολογίζει το γενικό μέσο όρο βαθμολογίας για όλη την τάξη της Γ΄ Λυκείου.

```
Αλγόριθμος Μέσος_Όρος
Σ ←0
Για i από 1 μέχρι 105
   Διάβασε Μ[ i ]
   Σ ←Σ+Μ[ i ]
Τέλος_επανάληψης
ΜΟ ←Σ/105
Αποτελέσματα // ΜΟ //
Τέλος Μέσος_Όρος
```

Έστω ότι δίνονται δύο δισδιάστατοι πίνακες Α και Β διαστάσεων 5Χ5 ο καθένας. Να γραφεί αλγόριθμος που θα διαβάζει τα στοιχεία των πινάκων και θα υπολογίζει το άθροισμά τους το οποίο θα αποθηκεύεται σε ένα νέο πίνακα.

```
Αλγόριθμος Άθροισμα_Πινάκων
Για i από 1 μέχρι 5
   Για j από 1 μέχρι 5
   Διάβασε A[i,j], B[i,j]
   C[i,j] ← A[i,j] + B[i,j]
    Τέλος_επανάληψης
Τέλος_επανάληψης
Αποτελέσματα // C //
Τέλος Μέσος_Όρος
```

Ο αριθμός των συνδυασμών Ν πραγμάτων ανά Κ ομάδες δίνεται από τον τύπο Ν!/ Κ!(Ν-Κ)!
π.χ. αν Ν=10 πράγματα ανά ομάδες των Κ=5

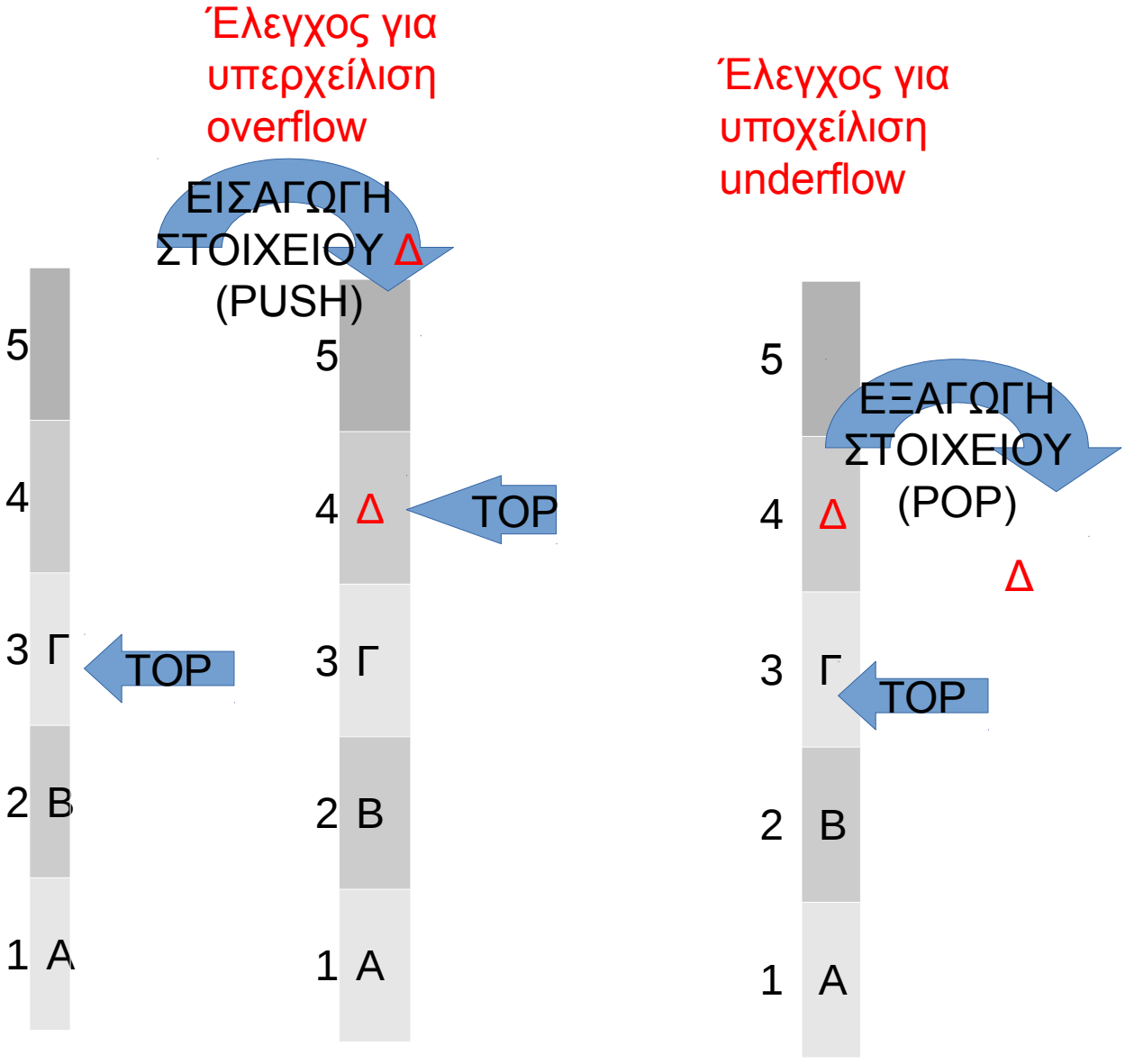
$$\frac{10!}{5!(10-5)!} = \frac{10X9X8X7X6X\cancel{5X4X2X1}}{5X4X3X2X1X\cancel{5X4X3X2X1}}$$

```
Αλγόριθμος Συνδυασμοί
Διάβασε Ν,Κ
α ← 1
Για i από Ν μέχρι Ν-Κ+1 Με_βήμα -1
   α←α * i
Τέλος_επανάληψης
β ← 1
Για i από 1 μέχρι Κ
   β←β * i
Τέλος_επανάληψης

Αποτελέσματα // α / β //
Τέλος Συνδυασμοί
```

3.4 Στοίβα

Last In First Out



Σε μία στοίβα εισάγονται μία μόνο φορά τρία στοιχεία τα Α,Β,Γ με τη σειρά που αναφέρονται, τα οποία αργότερα εξάγονται. Όμως οι εξαγωγές μπορούν να συμβούν οποτεδήποτε μεταξύ ή μετά τις εισαγωγές. Πόσες και ποιες είναι όλες οι πιθανές διατάξεις - αλληλοδιαδοχές στοιχείων (Α,Β,Γ) που μπορούμε να πάρουμε;

Το πλήθος των διατάξεων 3 στοιχείων είναι 3!=6

Συγκεκριμένα **ΑΒΓ, ΑΓΒ, ΒΑΓ, ΒΓΑ, ΓΒΑ και ΓΑΒ**

Η ΓΑΒ όμως δεν γίνεται αφού όταν εισάγονται τα στοιχεία στη στοίβα είναι πάντα με τη σειρά **ΑΒΓ**.

1: Εισ. Α, Εξ Α, Εισ. Β, Εξ Β, Εισ. Γ, Εξ Γ : **ΑΒΓ**

2: Εισ. Α, Εξ Α, Εισ. Β, Εισ. Γ, Εξ Γ, Εξ Β : **ΑΓΒ**

3: Εισ. Α, Εισ. Β, Εξ Β, Εξ Α, Εισ. Γ, Εξ Γ : **ΒΑΓ**

4: Εισ. Α, Εισ. Β, Εξ Β, Εισ. Γ, Εξ Γ, Εξ Α : **ΒΓΑ**

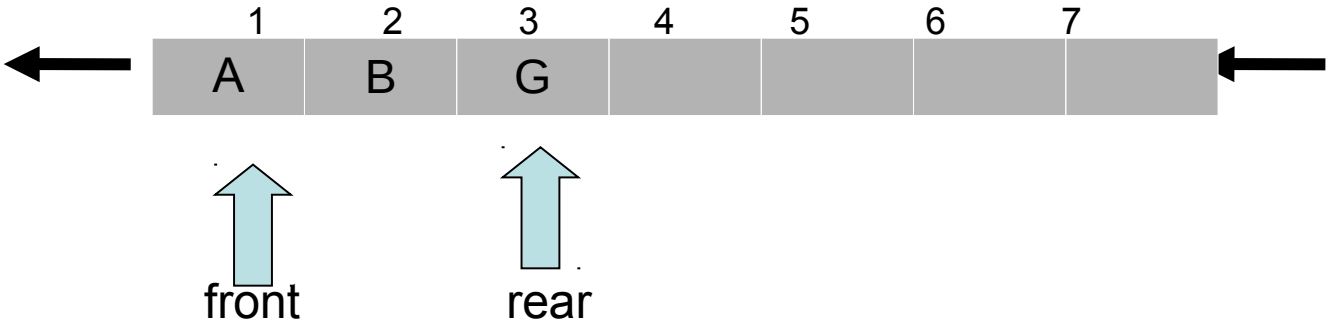
5: Εισ. Α, Εισ. Β, Εισ. Γ, Εξ Γ, Εξ Β, Εξ Α : **ΓΒΑ**

Η απάντηση τελικά είναι 5 διατάξεις :**ΑΒΓ , ΑΓΒ, ΒΑΓ, ΒΓΑ, ΓΒΑ**

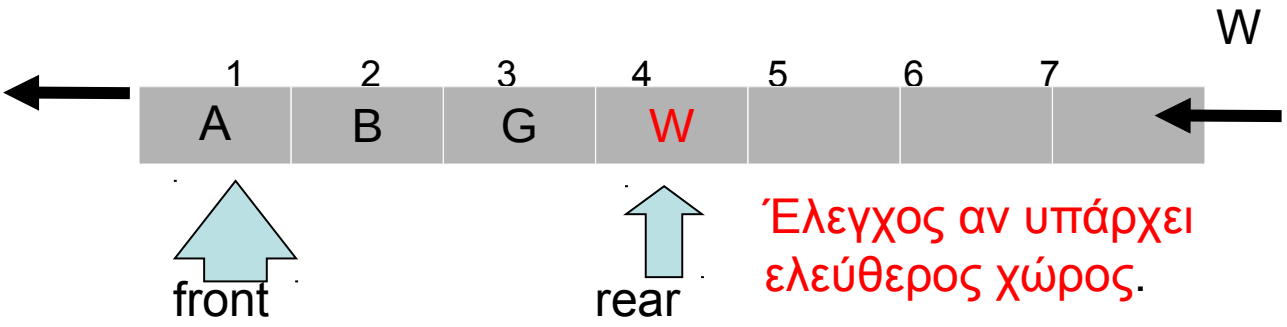
3.5 Ουρά

First In First Out

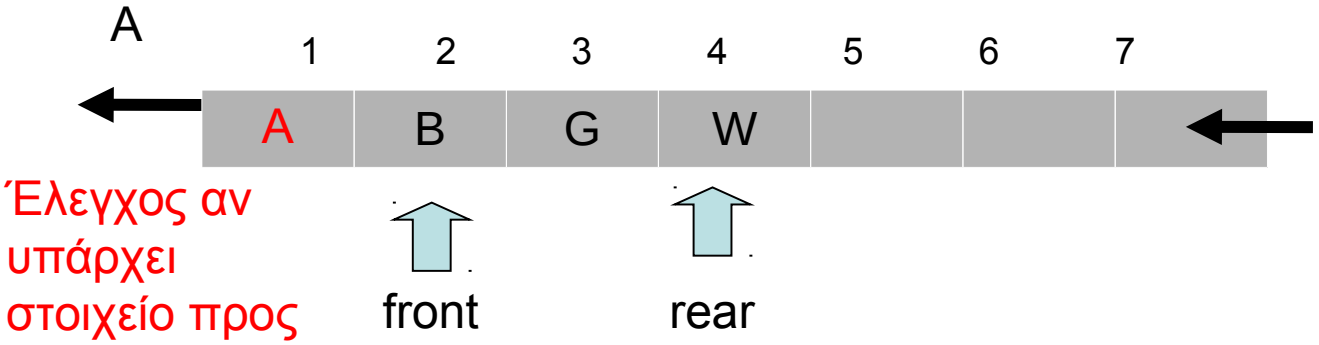
Η Ουρά με τρία στοιχεία



Λειτουργία εισαγωγής (enqueue)



Λειτουργία εξαγωγής (dequeue)



ΔΤ4_σελ34_τετ_μαθ:
Να δοθούν οι αλγόριθμοι εισαγωγής και εξαγωγής μιας ουράς μεγέθους 100 στοιχείων

Αλγόριθμος Εισαγωγή_στοιχείου_σε_ουρά
Δεδομένα //item,queue,rear//
Αν rear<100 **Τότε**
    rear ← rear+1
    queue [rear] ←item
    done ←Αληθής
Αλλιώς
    done ← Ψευδής
Τέλος_Αν
Αποτελέσματα //rear,done,queue//
Τέλος Εισαγωγή_στοιχείου_σε_ουρά

Αλγόριθμος Εξαγωγή_στοιχείου_από_ουρά
Δεδομένα //queue,front,rear//
Αν rear ≥ front **Τότε**
    item ← queue[front]
    front ←front+1
    done ←Αληθής
Αλλιώς
    done ← Ψευδής
Τέλος_Αν
Αποτελέσματα //item,done,front//
Τέλος Εισαγωγή_στοιχείου_σε_ουρά

3.6 Σειριακή Αναζήτηση.

1	3	5	-1	123	6	38

Αλγόριθμος Sequential_search
Δεδομένα //n, table, key//! n μέγεθος πίνακα table
! key στοιχείο προς αναζήτηση
! table μονοδιάστατος αταξινόμητος πίνακας
! ακέραιων αριθμών
done ← ψευδής
position ← 0
i ← 1
Όσο (done=ψευδής) και (i≤n) Επανάλαβε
    Αν table[i] = key τότε
      done ← αληθής
      position ← i
    αλλιώς *! *Η περίπτωση αλλιώς δεν χρειάζεται*
      i ← i +1
    Τέλος_αν
Τέλος_επανάληψης
Αποτελέσματα //done, position//
Τέλος Sequential_search

3.6 Σειριακή Αναζήτηση.

1	3	5	-1	5	6	3

```
Αλγόριθμος Sequential_search
Δεδομένα //n, table, key//! n μέγεθος πίνακα table
! key στοιχείο προς αναζήτηση
! table μονοδιάστατος αταξινόμητος πίνακας
! ακέραιων αριθμών
position ← 0
i ← 1
Όσο i≤n Επανάλαβε
    Αν table[i] = key τότε
        Εμφάνισε ¨Υπάρχει στη θέση:¨,i
        position ← i
    Τέλος_αν
    i ← i +1
Τέλος_επανάληψης
Αν position = 0 τότε Εμφάνισε ¨ΤΟ ΣΤΟΙΧΕΙΟ ΔΕΝ ΥΠΑΡΧΕΙ¨
Τέλος Sequential_search
```

3.6 Σειριακή Αναζήτηση.

-1	1	3	3	5	5	6

```
Αλγόριθμος Sequential_search
Δεδομένα //n, table, key//! n μέγεθος πίνακα table
! key στοιχείο προς αναζήτηση
! table μονοδιάστατος ταξινομημένος πίνακας
! ακέραιων αριθμών
position ← 0
i ← 1
Όσο i≤n   Επανάλαβε
    Αν table[i] = key τότε
        Εμφάνισε ¨Υπάρχει στη θέση:¨,i
        position ← I
    Αλλιώς
        ΑΝ key<table[i] τότε
            i ← n +1
        Τέλος_αν
    Τέλος_αν
    i ← i +1
Τέλος_επανάληψης
Αν position = 0 τότε Εμφάνισε ¨ΤΟ ΣΤΟΙΧΕΙΟ ΔΕΝ ΥΠΑΡΧΕΙ¨
Τέλος Sequential_search
```

Αν ο πίνακας είναι ταξινομημένος πότε σταματάει η σειριακή αναζήτηση.

3.6 Σειριακή Αναζήτηση.

Είναι η απλούστερη μέθοδος αναζήτησης αλλά και η λιγότερο αποτελεσματική.

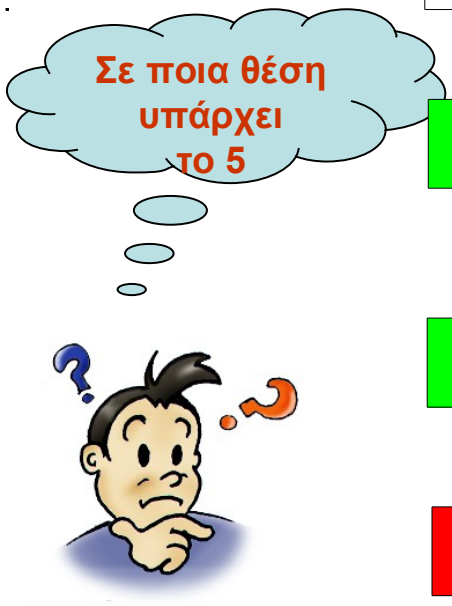
Χρησιμοποιείται συνήθως:

Όταν ο πίνακας είναι αταξινόμητος.

Όταν ο πίνακας είναι μικρού μεγέθους.

Όταν η αναζήτηση στον συγκεκριμένο πίνακα γίνεται σπάνια.

3.6 Δυαδική Αναζήτηση.

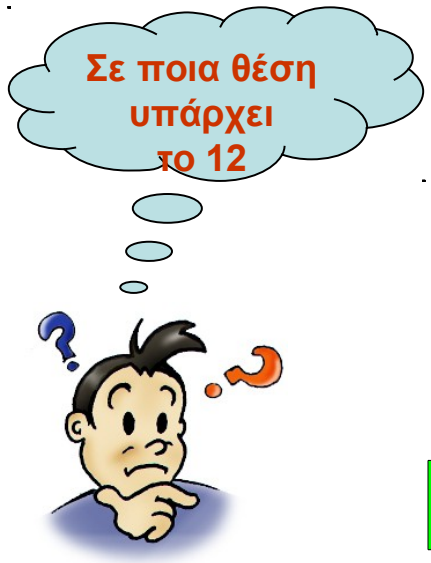


A	1	3	5	13	13	36	38

S			I			E
1	3	5	13	13	36	38

S	I	E				
1	3	5	13	13	36	38

		SIE				
1	3	5	13	13	36	38



			A			
1	3	5	13	13	36	38

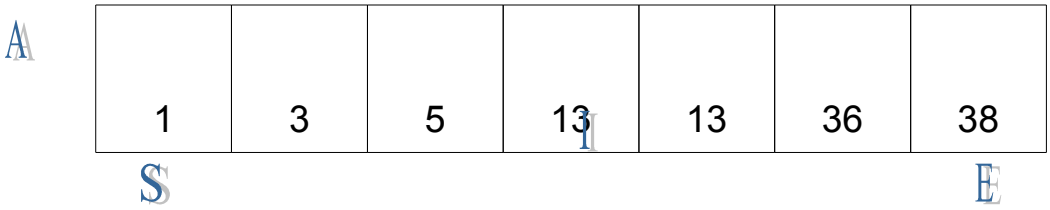
S			I			E
1	3	5	13	13	36	38

S	I	E				
1	3	5	13	13	36	38

		SIE				
1	3	5	13	13	36	38

		E	S			
1	3	5	13	13	36	38

3.6 Δυαδική Αναζήτηση.



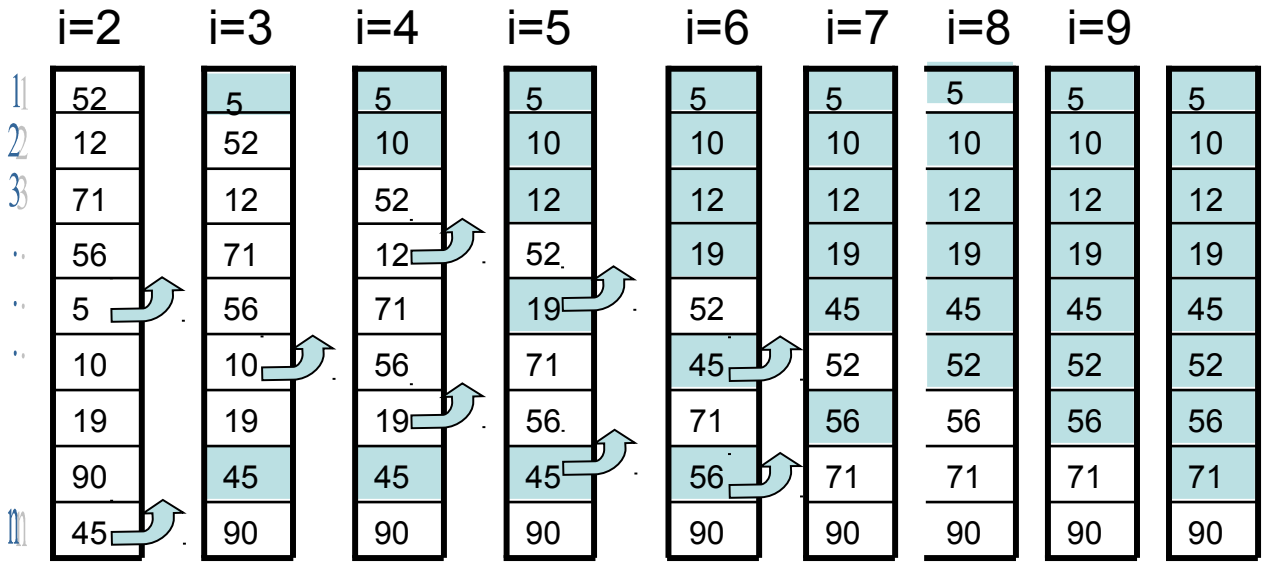

```
ΑΛΓΟΡΙΘΜΟΣ Δυαδική_αναζήτηση
ΔΕΔΟΜΕΝΑ // A, N // ! Ο πίνακας Α με Ν ταξινομημένα κατά αύξουσα σειρά
         !στοιχεία
ΔΙΑΒΑΣΕ X  !Το Χ στοιχείο προς αναζήτηση
S ← 1
E ← N
ΘΕΣΗ ← 0 ! Περιέχει τη θέση του Χ στον Α. Αν είναι 0 σημαίνει ότι δεν υπάρχει
I ← 0          !  στον Α
ΟΣΟ S ≤ E ΕΠΑΝΑΛΑΒΕ

        I ← (S + E) / 2 !Θεώρησε μόνο το ακέραιο μέρος της διαίρεσης
        ΑΝ X=A[ I ] ΤΟΤΕ
           ΘΕΣΗ ← I
           S ← E+1
        ΑΛΛΙΩΣ_ ΑΝ X<A[ I ] ΤΟΤΕ
              E ← I – 1
         ΑΛΛΙΩΣ
              S ← I + 1
        ΤΕΛΟΣ_ΑΝ


ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΠΟΤΕΛΕΣΜΑΤΑ //ΘΕΣΗ//
ΤΕΛΟΣ Δυαδική_αναζήτηση
```

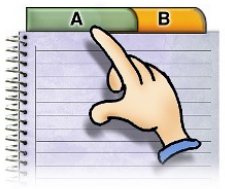
3.7 ΤΑΞΙΝΟΜΗΣΗ

Αλγόριθμος Φυσσαλίδας

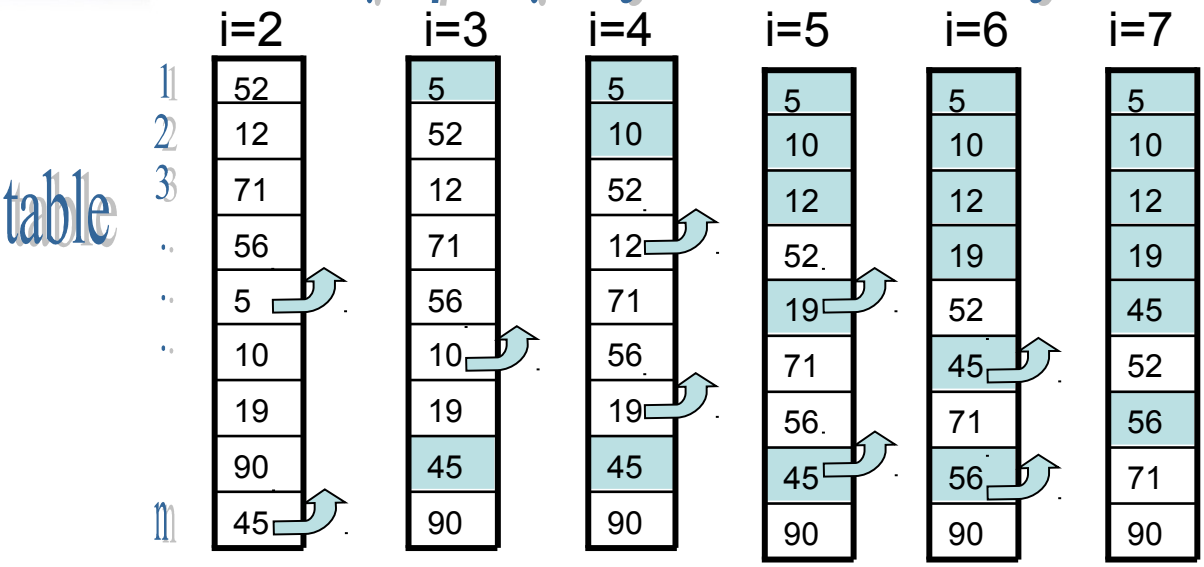



```
Αλγόριθμος Φυσσαλίδα
Δεδομένα  // table, n //
Για i από 2 μέχρι n
        Για j από n μέχρι i με_βήμα -1
                Αν table[ j -1] > table[ j ] τότε
                    αντιμετάθεσε table[ j -1 ], table[ j ]
                Τέλος_αν
        Τέλος_επανάληψης
Τέλος_επανάληψης
Αποτέλεσμα // table //
Τέλος Φυσσαλίδα
```

3.7 ΤΑΞΙΝΟΜΗΣΗ



Αλγόριθμος Φυσσαλίδας




```
Αλγόριθμος Φυσσαλίδα_ΔΤ2_σελ33
Δεδομένα  // table, n //
i ←2
Αρχή_επανάληψης
      flag ←0
      Για j από n μέχρι i με_βήμα -1
            Αν table[ j -1] > table[ j ] τότε
                  m ← table[ j -1 ] !Η εντολή αντιμετάθεσης τιμών με
                   table[j-1] ← table[ j ] ! χρήση μίας βοηθητικής
                   table[j] ←m     !μεταβλητής
                   flag ←1
            Τέλος_αν
      Τέλος_επανάληψης
       i ←i+1
Μέχρις_ότου (flag = 0) ή (i>n)
Αποτέλεσμα // table //
Τέλος Φυσσαλίδα_ΔΤ2_σελ33
```

3.7 ΤΑΞΙΝΟΜΗΣΗ

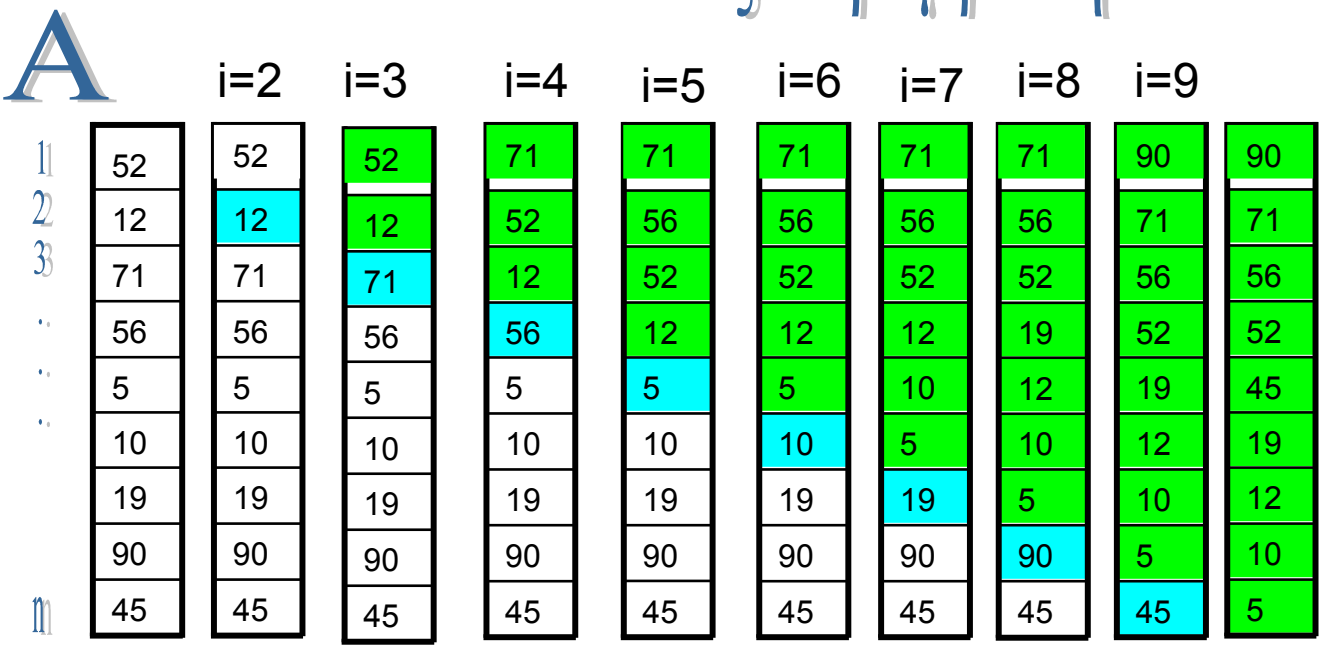
Με Επιλογή

table	i=1	i=2	i=3	i=4	i=5	i=6	i=7	i=8	
1	52	5	5	5	5	5	5	5	5
2	12	12	10	10	10	10	10	10	10
3	71	71	71	12	12	12	12	12	12
..	56	56	56	56	19	19	19	19	19
..	5	52	52	52	52	45	45	45	45
..	10	10	12	71	71	71	52	52	52
	19	19	19	19	56	56	56	56	56
	90	90	90	90	90	90	90	90	71
n	45	45	45	45	45	52	71	71	90

```
Αλγόριθμος Ταξινόμηση_με_επιλογή
Δεδομένα  // table, n //
Για i από 1 μέχρι n-1
    k← i
    x←table[ i ]
    Για j από i+1 μέχρι n
            Αν x > table[ j ] τότε
                x← table[ j ]
                k ← j
            τέλος_αν
        τέλος_επανάληψης
   table[ k ]← table[ i ]
   table[ i ]← x
τέλος_επανάληψης
Αποτέλεσμα // table //
Τέλος Ταξινόμηση_με_επιλογή
```

3.7 ΤΑΞΙΝΟΜΗΣΗ

Με απ`ευθείας παρεμβολή




```
Αλγόριθμος Ταξινόμηση_με_ευθεία_παρεμβολή
Δεδομένα  // A, n //
Για i από 2 μέχρι n
    χ ← A[ i ]
    κ ← i
    t ←ΑΛΗΘΗΣ
    Οσο  t  ΚΑΙ κ > 1 Επανάλαβε
        ΑΝ χ > A[κ - 1] ΤΟΤΕ
            A[κ] ← A[κ - 1]
             κ ← κ - 1
        ΑΛΛΙΩΣ
            t ←ΨΕΥΔΗΣ
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    A[κ] ← χ
τέλος_επανάληψης
Αποτέλεσμα // A //
Τέλος Ταξινόμηση_με_ευθεία_παρεμβολή
```

3.9 Άλλες δομές δεδομένων

Οι δομές δεδομένων που χρησιμοποιούν δείκτες έτσι ώστε να μην απαιτείται εκ των προτέρων καθορισμός του μέγιστου αριθμού κόμβων ονομάζονται δυναμικές. Οι κόμβοι σ`αυτές δεν είναι απαραίτητα σε συνεχόμενες θέσης της μνήμης.

Λίστες

Δένδρα

Γράφοι

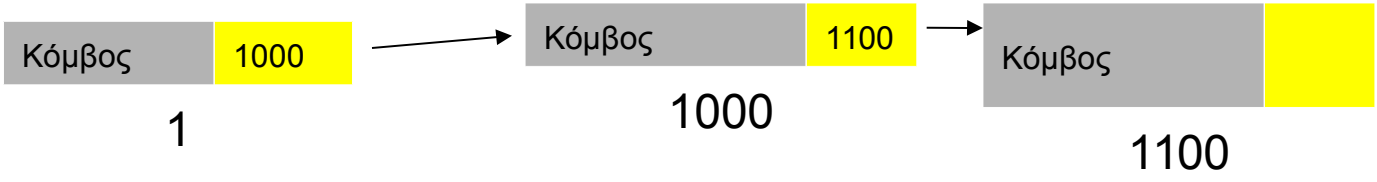
3.9 Άλλες δομές δεδομένων
Λίστα

Ο δείκτης στις γλώσσες προγραμματισμού είναι ένας ιδιαίτερος τύπος δεδομένων με τιμές διευθύνσεις της κύριας μνήμης

ΚΟΜΒΟΣ ΛΙΣΤΑΣ

Δεδομένα	Δείκτης

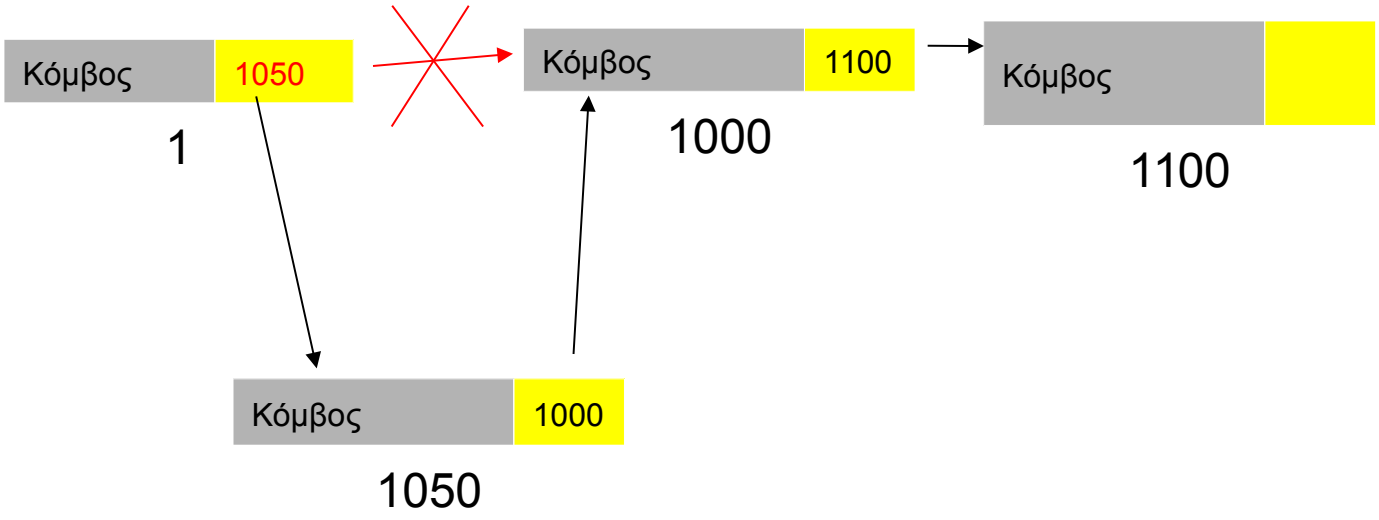
Μία ή περισσότερες αριθμητικές ή αλφαριθμητικές πληροφορίες



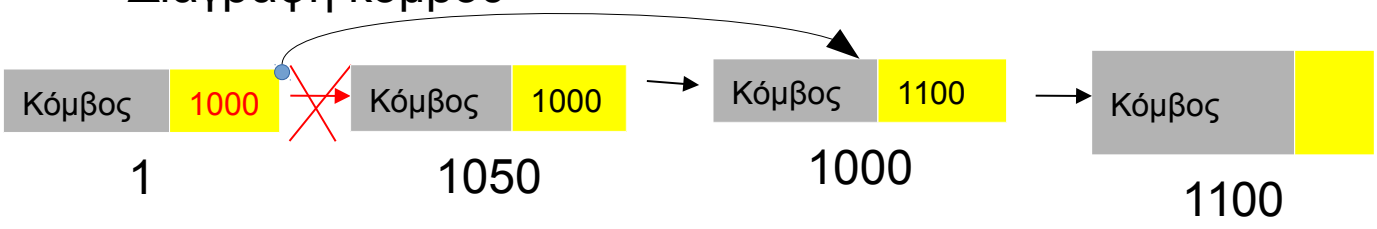
1 1000 1100

3.9 Άλλες δομές δεδομένων
Λίστα

Εισαγωγή – παρεμβολή νέου κόμβου



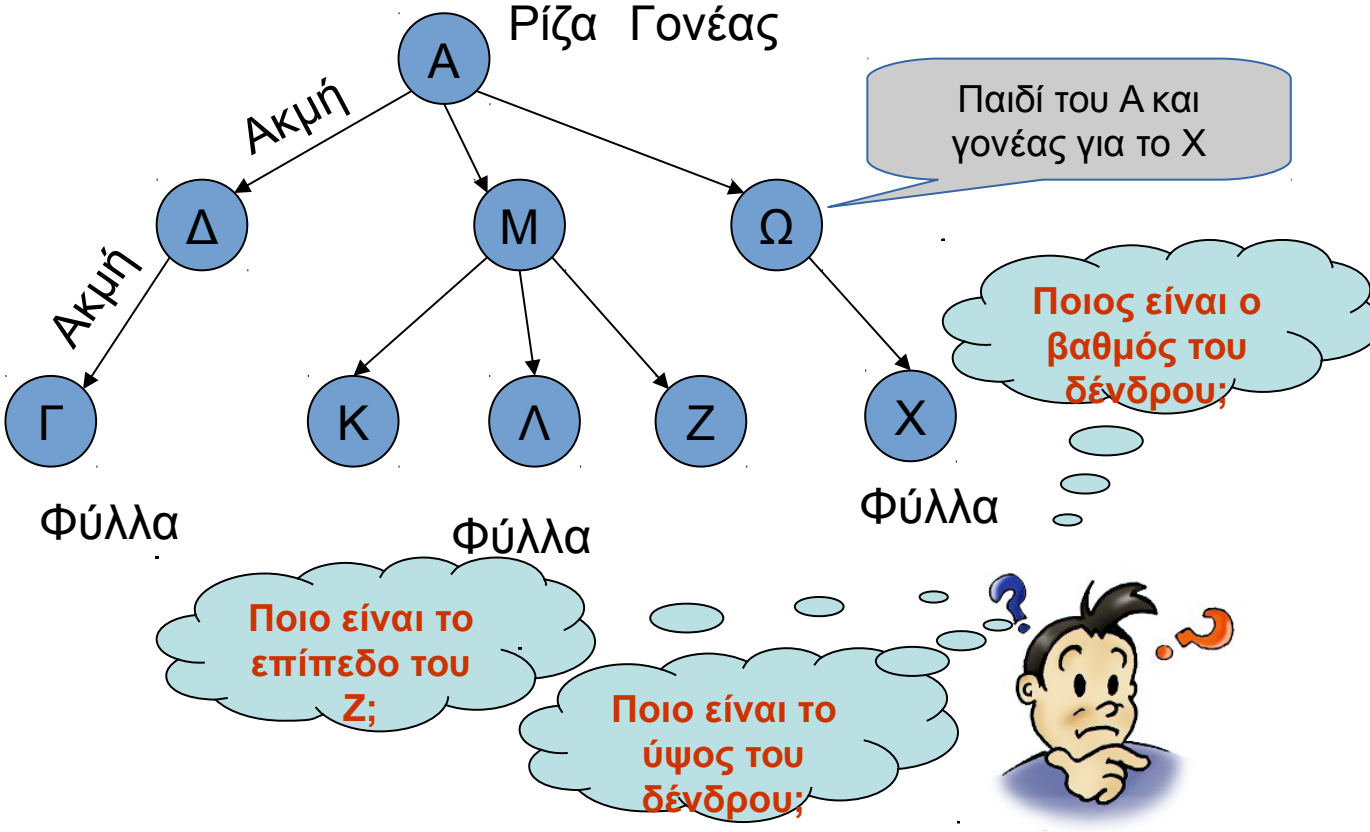
Διαγραφή κόμβου



3.9 Άλλες δομές δεδομένων
Δένδρα

Δένδρο είναι ένα πεπερασμένο σύνολο κόμβων (ίδιου τύπου) και ακμών που συνδέουν τους κόμβους με βάση κάποια ιεραρχική σχέση.

- Βαθμός κόμβου είναι ο αριθμός των παιδιών του.
- Βαθμός δένδρου είναι ο μέγιστος βαθμός από όλους τους κόμβους.
- Φύλλα ή τερματικοί είναι οι κόμβοι από τους οποίους δεν ξεκινάνε ακμές.
- Επίπεδο κόμβου είναι το πλήθος των προγόνων του. Η ρίζα έχει επίπεδο 0
- Ύψος δένδρου είναι το μέγιστο επίπεδο από όλους τους κόμβους.



3.9 Άλλες δομές δεδομένων
Γράφοι

Γράφος είναι ένα πεπερασμένο σύνολο κόμβων (ίδιου τύπου) και ακμών που συνδέουν τους κόμβους χωρίς κάποια ιεραρχία. Αποτελούν την ποιο γενική μορφή δομής δεδομένων, επομένως οι προηγούμενες δομές είναι υποπεριπτώσεις των γράφων.