

ΤαξινόμησηΤαξινόμηση ευθείας ανταλλαγής (Φυσαλίδα)

1) Να ταξινομηθεί ο πίνακας A[N] σε αύξουσα σειρά με τη μέθοδο της **φυσαλίδας**.

```

Για i από 2 μέχρι N
  Για j από N μέχρι i με_βήμα -1
    Αν (A[j] < A[j-1]) τότε
      tmp <-- A[j]
      A[j] <-- A[j-1]
      A[j-1] <-- tmp
    Τέλος_Αν
  Τέλος_Επανάληψης
Τέλος_Επανάληψης

```

2) Να ταξινομηθεί ο πίνακας B[N] σε φθίνουσα σειρά διατηρώντας την αντιστοιχία με τον πίνακα ON[N]. Σε περίπτωση ισοβαθμίας δύο ή περισσότερων στοιχείων στον πίνακα B[N], τότε να εμφανίζονται τα αντίστοιχα στοιχεία του ON[N] με αλφαβητική σειρά.

```

Για i από 2 μέχρι N
  Για j από N μέχρι i με_βήμα -1
    Αν (B[j] > B[j-1]) τότε
      tmp1 <-- B[j]           ! Διαφορετικό tmp για κάθε
      B[j] <-- B[j-1]       ! πίνακα διαφορετικού τύπου
      B[j-1] <-- tmp1       ! δεδομένων.

      tmp2 <-- ON[j]
      ON[j] <-- ON[j-1]
      ON[j-1] <-- tmp2
    Αλλιώς_αν (B[j] = B[j-1]) τότε
      Αν (ON[j] < ON[j-1]) τότε
        tmp2 <-- ON[j]     ! Οι βαθμοί δεν χρειάζονται
        ON[j] <-- ON[j-1]  ! αντιμετάθεση, καθώς είναι
        ON[j-1] <-- tmp2   ! ίσοι.
      Τέλος_Αν
    Τέλος_Αν
  Τέλος_Επανάληψης
Τέλος_Επανάληψης

```

3) Να ταξινομηθεί ο πίνακας B[N] σε αύξουσα σειρά. Σε περίπτωση που η ταξινόμηση έχει ολοκληρωθεί σε λιγότερες (από N-1) σαρώσεις, τότε η φυσαλίδα να σταματάει (έξυπνη φυσαλίδα).

```

Αρχή_Επανάληψης
  πλ <-- 0           ! πλ = πλήθος αντιμεταθέσεων κάθε
  i <- 2            ! σάρωσης. Εάν παραμείνει 0 μετά
  Για j από N μέχρι i με_βήμα -1 ! από κάποια σάρωση, τότε ο
    Αν (A[j] < A[j-1]) τότε      ! πίνακας είναι πλέον
      πλ <-- πλ + 1           ! ταξινομημένος.

      tmp <-- A[j]
      A[j] <-- A[j-1]
      A[j-1] <-- tmp
    Τέλος_Αν
  Τέλος_Επανάληψης
  i <-- i + 1       ! Η φυσαλίδα τελειώνει, είτε όταν
Μέχρις_Ότου (i > N Η πλ = 0)   ! τελειώσουν οι σαρώσεις, είτε
                                ! στην πρώτη σάρωση που θα κάνει
                                ! μηδέν (0) αντιμεταθέσεις.

```

4) Να ταξινομηθεί ο διδιάστατος πίνακας $A[M, N]$ σε αύξουσα σειρά.

a. Μεταφορά όλων των στοιχείων του διδιάστατου πίνακα $A[M,N]$ σε μονοδιάστατο πίνακα B , μεγέθους $M*N$

```
κ <-- 1
Για i από 1 μέχρι M
  Για j από 1 μέχρι N
    B[κ] <-- A[i,j]
    κ <-- κ + 1
  Τέλος_Επανάληψης
Τέλος_Επανάληψης
```

b. Ταξινόμηση του μονοδιάστατου πίνακα B (με μέγεθος $M*N$)

```
Για i από 2 μέχρι M*N
  Για j από M*N μέχρι i με_βήμα -1
    Αν (B[j] < B[j-1]) τότε
      tmp <-- B[j]
      B[j] <-- B[j-1]
      B[j-1] <-- tmp
    Τέλος_Αν
  Τέλος_Επανάληψης
Τέλος_Επανάληψης
```

c. Μεταφορά όλων των στοιχείων του μονοδιάστατου πίνακα B ξανά πίσω στο διδιάστατο πίνακα $A[M,N]$

```
κ <-- 1
Για i από 1 μέχρι M
  Για j από 1 μέχρι N
    A[i,j] <-- B[κ]
    κ <-- κ + 1
  Τέλος_Επανάληψης
Τέλος_Επανάληψης
```

Ταξινόμηση

5) Να ταξινομηθεί η **γραμμή i** του διδιάστατου πίνακα $A[M, N]$ σε αύξουσα σειρά.

```
Για κ από 2 μέχρι N
  Για j από N μέχρι i με_βήμα -1
    Αν (A[i, j] < A[i, j-1]) τότε
      tmp <-- A[i, j]
      A[i, j] <-- A[i, j-1]
      A[i, j-1] <-- tmp
  Τέλος_Αν
Τέλος_Επανάληψης
Τέλος_Επανάληψης
```

6) Να ταξινομηθεί κάθε **γραμμή** του διδιάστατου πίνακα $A[M, N]$ σε αύξουσα σειρά.

Για i από 1 μέχρι M

```
Για κ από 2 μέχρι N
  Για j από N μέχρι i με_βήμα -1
    Αν (A[i, j] < A[i, j-1]) τότε
      tmp <-- A[i, j]
      A[i, j] <-- A[i, j-1]
      A[i, j-1] <-- tmp
  Τέλος_Αν
Τέλος_Επανάληψης
Τέλος_Επανάληψης
```

Τέλος_Επανάληψης

Ταξινόμηση με επιλογή

1) Να ταξινομηθεί ο πίνακας $B[N]$ σε αύξουσα σειρά με τη μέθοδο της **επιλογής**.

Για i από 1 μέχρι $N - 1$

```
min ← B[i]
θmin ← i
Για j από i+1 μέχρι N
    Αν (B[j] < min) τότε
        min ← B[j]
        θmin ← j
Τέλος_ΑΝ
Τέλος_Επανάληψης
```

```
tmp ← B[θmin]
B[θmin] ← B[i]
B[i] ← tmp
```

Τέλος_Επανάληψης

2) Παραλλαγή χωρίς τη χρήση της μεταβλητής **min**.

Για i από 1 μέχρι $N - 1$

```
θmin ← i
Για j από i+1 μέχρι N
    Αν (B[j] < B[θ]) τότε
        θmin ← j
Τέλος_ΑΝ
Τέλος_Επανάληψης
```

```
tmp1 ← B[θmin]
B[θmin] ← B[i]
B[i] ← tmp1
```

Τέλος_Επανάληψης

3) Να ταξινομηθεί ο πίνακας $B[N]$ σε φθίνουσα σειρά διατηρώντας την αντιστοιχία με τον πίνακα $ON[N]$.

Η μέθοδος είναι ακριβώς ίδια με τις προηγούμενες. Απλά κάθε φορά που αντιμεταθέτουμε κάποιο στοιχείο του B , πρέπει να κάνουμε την ίδια αντιμετάθεση (χωρίς να ρωτήσουμε) και στον παράλληλο πίνακα ON .

```
tmp2 ← ON[θmin]
ON[θmin] ← ON[i]
ON[i] ← tmp2
```

Ταξινόμηση

4) Να ταξινομηθεί ο πίνακας $B[N]$ σε φθίνουσα σειρά διατηρώντας την αντιστοιχία με τον πίνακα $ON[N]$. Σε περίπτωση ισοβαθμίας δύο ή περισσότερων στοιχείων στον πίνακα $B[N]$, τότε να εμφανίζονται τα αντίστοιχα στοιχεία του $ON[N]$ με αλφαβητική σειρά.

Για i από 1 μέχρι $N - 1$

```
min ← B[i]
θmin ← i
Για j από i+1 μέχρι N
  Αν (B[j] < min) τότε
    min ← B[j]
    θmin ← j
  Αλλιώς_αν (B[j] = min) τότε      ! Αν ξαναβρούμε στοιχείο ίσο
  Αν (ON[j] < ON[θmin]) τότε      ! με το min, τότε κρατάμε
  θmin ← j                        ! τη δική του θέση, εφόσον
  Τέλος_αν                          ! το όνομα σε αυτήν την θέση
  Τέλος_ΑΝ                          ! είναι μικρότερο από το
Τέλος_Επανάληψης                  ! όνομα στην τρέχουσα θmin.

tmp1 ← B[θmin]
B[θmin] ← B[i]
B[i] ← tmp1

tmp2 ← ON[θmin]
ON[θmin] ← ON[i]
ON[i] ← tmp2
```

Τέλος_Επανάληψης

Ταξινόμηση με παρεμβολή

1) Να ταξινομηθεί ο πίνακας $B[N]$ σε αύξουσα σειρά με τη μέθοδο της **παρεμβολής**.

Σημείωση:

Η μέθοδος είναι πιο εύκολα κατανοητή όταν φανταστούμε ότι πρέπει να παρεμβάλλουμε το **στοιχείο i** του πίνακα B , στον ήδη ταξινομημένο (από τις προηγούμενες σαρώσεις) υπό-πίνακα $B[1$ έως $i-1]$.

Για i από 2 μέχρι N

```
key ← B[i]           ! key = το στοιχείο που θέλουμε να παρεμβάλλουμε
                    ! στον ήδη ταξινομημένο πίνακα B[1 έως i-1]
```

```
stop ← Ψευδής
j ← i - 1
Όσο (j >= 1) και (stop = Ψευδής) επανάλαβε
  Αν (key < B[j]) τότε
    B[j+1] ← B[j]
    j ← j - 1
  Αλλιώς
    stop ← Αληθής
  Τέλος_Αν
Τέλος_Επανάληψης
```

```
B[j+1] ← key
```

Τέλος_Επανάληψης

2) Να ταξινομηθεί ο πίνακας $B[N]$ σε φθίνουσα σειρά διατηρώντας την αντιστοιχία με τον πίνακα $ON[N]$.

Η μέθοδος είναι ακριβώς ίδια με την παραπάνω. Απλά κάθε φορά που ένα στοιχείο του B μετακινείται κατά μία θέση προς τα δεξιά, πρέπει να κάνουμε την ίδια μετακίνηση (χωρίς να ρωτήσουμε) και στον παράλληλο πίνακα ON .

Για i από 2 μέχρι N

```
key1 ← B[i]         ! key1 = το στοιχείο του B που θα παρεμβάλλουμε
key2 ← ON[i]       ! key2 = το στοιχείο του ON που θα παρεμβάλλουμε
```

```
stop ← Ψευδής
j ← i - 1
Όσο (j >= 1) και (stop = Ψευδής) επανάλαβε
  Αν (key1 < B[j]) τότε
    B[j+1] ← B[j]
    ON[j+1] ← ON[j]
    j ← j - 1
  Αλλιώς
    stop ← Αληθής
  Τέλος_Αν
Τέλος_Επανάληψης
```

```
B[j+1] ← key1
ON[j+1] ← key2
```

Τέλος_Επανάληψης

Ταξινόμηση

3) Να ταξινομηθεί ο πίνακας $B[N]$ σε φθίνουσα σειρά διατηρώντας την αντιστοιχία με τον πίνακα $ON[N]$. Σε περίπτωση ισοβαθμίας δύο ή περισσότερων στοιχείων στον πίνακα $B[N]$, τότε να εμφανίζονται τα αντίστοιχα στοιχεία του $ON[N]$ με αλφαβητική σειρά.

Για i από 2 μέχρι N

```
key1 ← B[i]      ! key1 = το στοιχείο του B που θα παρεμβάλουμε  
key2 ← ON[i]     ! key2 = το στοιχείο του ON που θα παρεμβάλουμε
```

```
stop ← Ψευδής  
j ← i - 1  
Όσο (j >= 1) και (stop = Ψευδής) επανάλαβε  
  Αν (key1 < B[j]) τότε  
    B[j+1] ← B[j]  
    ON[j+1] ← ON[j]  
    j ← j - 1  
  Αλλιώς_αν (key1 = B[j]) τότε  
    Αν (key2 < ON[j]) τότε  
      ON[j+1] ← ON[j]  
    Τέλος_Αν  
  Αλλιώς  
    stop ← Αληθής  
  Τέλος_Αν  
Τέλος_Επανάληψης  
  
B[j+1] ← key1  
ON[j+1] ← key2
```

Τέλος_Επανάληψης

Αναζήτηση

1) Να γραφεί αλγόριθμος ο οποίος σε έναν μονοδιάστατο μη ταξινομημένο πίνακα A με μέγεθος N, θα αναζητά την πρώτη εμφάνιση του στοιχείου key. Σε περίπτωση που βρεθεί το στοιχείο key, η αναζήτηση πρέπει να τερματίζεται.

```

toBrika ← Ψευδής
i ← 1
Όσο (i ≤ N) και (toBrika = Ψευδής) επανέλαβε
  Αν (A[i] = key) τότε
    toBrika ← Αληθής
    θ ← i
  Αλλιώς
    i ← i + 1
  Τέλος_Αν
Τέλος_Επανάληψης

Αν (toBrika = Αληθής) τότε
  Γράψε 'Το στοιχείο', key, 'βρέθηκε στη θέση', θ
Αλλιώς
  Γράψε 'Το στοιχείο', key, 'δεν υπάρχει στον πίνακα'
Τέλος_Αν

```

Παραλλαγή:

2) Αν ζητηθεί η τελευταία εμφάνιση του στοιχείου key και όχι η πρώτη, τότε απλά ξεκινάμε τη σειριακή αναζήτηση από το τέλος προς την αρχή.

```

toBrika ← Ψευδής
i ← N
Όσο (i ≥ 1) και (toBrika = Ψευδής) επανέλαβε
  Αν (A[i] = key) τότε
    toBrika ← Αληθής
    θ ← i
  Αλλιώς
    i ← i - 1
  Τέλος_Αν
Τέλος_Επανάληψης

Αν (toBrika = Αληθής) τότε
  Γράψε 'Το στοιχείο', key, 'βρέθηκε στη θέση', θ
Αλλιώς
  Γράψε 'Το στοιχείο', key, 'δεν υπάρχει στον πίνακα'
Τέλος_Αν

```


Ταξινόμηση

3) Να γραφεί αλγόριθμος ο οποίος σε έναν μονοδιάστατο ταξινομημένο (με αύξουσα ταξινόμηση) πίνακα A με μέγεθος N, θα αναζητά την πρώτη εμφάνιση του στοιχείου key. Σε περίπτωση που βρεθεί το στοιχείο key, η αναζήτηση πρέπει να τερματίζεται.

Σημείωση:

Αυτό που αλλάζει είναι ότι, λόγω της ταξινόμησης, μπορούμε να σταματήσουμε την αναζήτηση, μόλις βρεθεί ένα στοιχείο μεγαλύτερο από αυτό που ψάχνουμε.

```
τοBrika ← Ψευδής
i ← 1
Όσο (i ≤ N) και (τοBrika = Ψευδής) επανέλαβε
  Αν (A[i] = key) τότε
    τοBrika ← Αληθής
    θ ← i
  Αλλιώς_αν (A[i] > key) τότε
    i ← N + 1                                ! Κάνοντας το i = N + 1
  Αλλιώς                                       ! τερματίζουμε πρόωρα την
    i ← i + 1                                  ! επανάληψη
  Τέλος_Αν
Τέλος_Επανάληψης

Αν (τοBrika = Αληθής) τότε
  Γράψε 'Το στοιχείο', key, 'βρέθηκε στη θέση', θ
Αλλιώς
  Γράψε 'Το στοιχείο', key, 'δεν υπάρχει στον πίνακα'
Τέλος_Αν
```

Παραλλαγή:

Αν ζητηθεί η τελευταία εμφάνιση του στοιχείου key και όχι η πρώτη, τότε πρέπει να αφαιρέσουμε από τη συνθήκη το τμήμα (τοBrika = Ψευδής) ώστε να συνεχιστεί η αναζήτηση ακόμη και αν βρεθεί το πρώτο στοιχείο. Δηλαδή:

```
τοBrika ← Ψευδής
i ← 1
Όσο (i ≤ N) επανέλαβε
  Αν (A[i] = key) τότε
    τοBrika ← Αληθής                        ! Όταν βρεθεί το στοιχείο
    θ ← i                                    ! δεν σταματάμε
    i ← i + 1
  Αλλιώς_αν (A[i] > key) τότε
    i ← N + 1                                ! Κάνοντας το i = N + 1
  Αλλιώς                                       ! τερματίζουμε πρόωρα την
    i ← i + 1                                  ! επανάληψη
  Τέλος_Αν
Τέλος_Επανάληψης

Αν (τοBrika = Αληθής) τότε
  Γράψε 'Το τελευταίο στοιχείο', key, 'βρέθηκε στη θέση', θ
Αλλιώς
  Γράψε 'Το στοιχείο',
```

Ερώτηση:

Αν ο πίνακας είναι ταξινομημένος με φθίνουσα σειρά, τότε τι πρέπει να αλλάξουμε;