

3.6 Αναζήτηση

Η αναζήτηση ενός στοιχείου σε πίνακα είναι μια από τις τυπικές επεξεργασίες. Με την αναζήτηση κάνουμε προσπέλαση ενός πίνακα μέχρι να εντοπίσουμε στοιχείο με συγκεκριμένη τιμή. Η θέση μέσα στον πίνακα όπου εντοπίζεται το στοιχείο μπορεί να αξιοποιηθεί και στην περίπτωση που εργαζόμαστε με παράλληλους πίνακες

Υπάρχουν αρκετοί αλγόριθμοι για την αναζήτηση στοιχείου σε πίνακα. Στην ύλη μας είναι οι εξής αλγόριθμοι:

1. της Σειριακής αναζήτησης
2. της Δυαδικής αναζήτησης (ο πίνακας πρέπει να είναι ταξινομημένος).

Σειριακή Αναζήτηση

Η σειριακή μέθοδος αναζήτησης είναι η πιο απλή, αλλά και η λιγότερη αποτελεσματική μέθοδος αναζήτησης. Έτσι, δικαιολογείται η χρήση της μόνο σε περιπτώσεις όπου:

- ο πίνακας είναι μη ταξινομημένος,
- ο πίνακας είναι μικρού μεγέθους (για παράδειγμα, $n \leq 20$),
- η αναζήτηση σε ένα συγκεκριμένο πίνακα γίνεται σπάνια,

Παρακάτω στα αριστερά, βλέπετε στον αλγόριθμο Sequential Search αναζητείται σειριακά αν υπάρχει το στοιχείο με τιμή key στο μη ταξινομημένο πίνακα table n θέσεων. Εάν υπάρχει και μόνο τότε επιστρέφει τη θέση position (δηλαδή, έναν αριθμό από 1 ως n) μέσα στον πίνακα table όπου εντοπίστηκε το στοιχείο με τιμή key. Μετά την εκτέλεση του αλγορίθμου η μεταβλητή position επιστρέφει την τιμή 0, αν η αναζήτηση είναι ανεπιτυχής.

1	2	3	4	5	6	7	8	9
52	12	71	56	5	10	19	90	45

Παρακάτω στη μεσαία στήλη, βλέπετε το πρόγραμμα Σειρ_Αναζητηση1 στο οποίο αναζητείται σειριακά αν υπάρχει το στοιχείο με τιμή key που δίνει ο χρήστης στο μη ταξινομημένο πίνακα table 9 θέσεων. Έστω τώρα ότι ο χρήστης αναζητά πχ το στοιχείο 56 που υπάρχει μέσα στον πίνακα, το πρόγραμμα θα εμφανίσει "Το στοιχείο είναι στη θέση 4". Αν όμως αναζητά πχ το στοιχείο 99 που δεν υπάρχει μέσα στον πίνακα, το πρόγραμμα θα εμφανίσει "Το στοιχείο δεν υπάρχει". Δώστε προσοχή στη χρήση λογικής μεταβλητής (της done) η οποία έχει τη λειτουργία να τερματίζει την αναζήτηση όταν γίνει ΑΛΗΘΗΣ, έτσι ώστε να μην χρειαστεί να συνεχίσει η σειριακή προσπέλαση του table[9]

Παρακάτω στα δεξιά, βλέπετε το πρόγραμμα Σειρ_Αναζητηση, το οποίο είναι μια παραλλαγή του Σειρ_Αναζητηση1. Η διαφορά του είναι ότι δεν κάνει χρήση λογικής μεταβλητής (της done), αλλά αξιοποιεί την τιμή της ακέραιης μεταβλητής position η οποία έχει τη λειτουργία να τερματίζει την αναζήτηση όταν γίνει μη μηδενική

```
Αλγόριθμος Sequential_Search
Δεδομένα // n, table, key //
done ← ψευδής
position ← 0
i ← 1
Όσο (done=ψευδής) και (i<=n) επανάλαβε
Αν table[i]=key τότε
    done ← αληθής
    position ← i
αλλιώς
    i ← i+1
Τέλος_αν
Τέλος_επανάληψης
Αποτελέσματα //done, position //
Τέλος Sequential_Search
```

```
ΠΡΟΓΡΑΜΜΑ Σειρ_Αναζητηση1
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: table[9],key,position,i
    ΛΟΓΙΚΕΣ: done
ΑΡΧΗ
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 9
    ΔΙΑΒΑΣΕ table[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
done <- ΨΕΥΔΗΣ
θέση <- 0
ΔΙΑΒΑΣΕ key
ΟΣΟ done = ΨΕΥΔΗΣ ΚΑΙ i<=9 ΕΠΑΝΑΛΑΒΕ
    ΑΝ table[i]=key ΤΟΤΕ
        done <- ΑΛΗΘΗΣ
        position <- i
    ΑΛΛΙΩΣ
        i <- i+1
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ done = ΑΛΗΘΗΣ ΤΟΤΕ
    ΓΡΑΨΕ 'Το στοιχείο είναι στη θέση', position
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'Το στοιχείο δεν υπάρχει'
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

```
ΠΡΟΓΡΑΜΜΑ Σειρ_Αναζητηση 2
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: table[9],key,position,i
ΑΡΧΗ
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 9
    ΔΙΑΒΑΣΕ table[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΔΙΑΒΑΣΕ key
position <- 0
i <- 1
ΟΣΟ position = 0 ΚΑΙ i<=9 ΕΠΑΝΑΛΑΒΕ
    ΑΝ table[i]=key ΤΟΤΕ
        position <- i
    ΑΛΛΙΩΣ
        i <- i+1
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ position <> 0 ΤΟΤΕ
    ΓΡΑΨΕ 'Το στοιχείο είναι στη θέση', position
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'Το στοιχείο δεν υπάρχει'
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Διαδικής αναζήτησης

Όταν ο πίνακας στον οποίο θέλουμε να αναζητήσουμε στοιχείο συγκεκριμένης τιμής είναι ταξινομημένος, ο αλγόριθμος Διαδικής Αναζήτησης είναι προτιμότερος από τη σειριακή, διότι χρειάζεται κάτω από τον μισό χρόνο για να υλοποιηθεί.

Η μέθοδος αυτή στηρίζεται στη συνεχή διαίρεση του πίνακα, σύμφωνα με τα ακόλουθα βήματα:

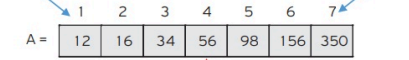
- Ελέγχουμε αν το ζητούμενο στοιχείο είναι ίσο με το στοιχείο του πίνακα που βρίσκεται στη μεσαία θέση.
- Αν όχι, ελέγχουμε αν το ζητούμενο στοιχείο είναι μικρότερο από το στοιχείο του πίνακα που βρίσκεται στη μεσαία θέση. Αν ισχύει, η αναζήτηση γίνεται μόνο στο 1ο μισό του πίνακα, διαφορετικά αλλάζουμε ψάχνουμε μόνο στο 2ο μισό του πίνακα
- Επαναλαμβάνονται τα προηγούμενα βήματα μέχρι να βρούμε το ζητούμενο στοιχείο.

Περίπτωση 1^η Θεωρούμε ότι τα στοιχεία του πίνακα A είναι ταξινομημένα κατά αύξουσα σειρά.

```

ΠΡΟΓΡΑΜΜΑ Διαδικής_Αναζήτηση1
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: A[7], KEY, τέλος, αρχ, μέση, i
  ΛΟΓΙΚΕΣ: δεικτης
ΑΡΧΗ
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 7
    ΔΙΑΒΑΣΕ A[i]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΡΑΨΕ 'Δώσε αριθμό για αναζήτηση στον πίνακα A.'
  ΔΙΑΒΑΣΕ KEY
  αρχ <- 1
  τέλος <- 7
  δεικτης <- ΨΕΥΔΗΣ
  ΟΣΟ αρχ <= τέλος ΚΑΙ δεικτης = ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ
    μέση <- (αρχ + τέλος) div 2
    ΑΝ μέση >= 1 ΚΑΙ μέση <= 7 ΤΟΤΕ
      ΑΝ KEY < A[μέση] ΤΟΤΕ
        τέλος <- μέση - 1
      ΑΛΛΙΩΣ_ΑΝ KEY > A[μέση] ΤΟΤΕ
        αρχ <- μέση + 1
      ΑΛΛΙΩΣ
        δεικτης <- ΑΛΗΘΗΣ
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΑΝ δεικτης = ΑΛΗΘΗΣ ΤΟΤΕ
    ΓΡΑΨΕ 'Βρέθηκε στη θέση ', μέση
  ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'Δε βρέθηκε'
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Διαδικής_Αναζήτηση1
    
```

Εστω ο πίνακας A[7], KEY= 34 και οι μεταβλητές αρχ, τέλος με την αρχικοποίηση

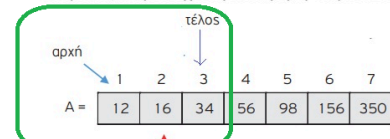


μέση

1η επανάληψη: μεταβλητή μέση παίρνει τιμή 4

Επειδή ισχύει $KEY < A[μέση]$ η μεταβλητή τέλος παίρνει τιμή 3

Επομένως, στη συνέχεια η αναζήτηση θα γίνει στο 1ο μισό

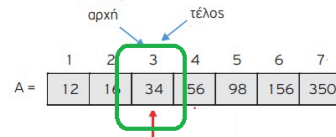


μέση

2η επανάληψη: μεταβλητή μέση παίρνει τιμή 2

Επειδή ισχύει $KEY > A[μέση]$ η μεταβλητή αρχ παίρνει τιμή 3

Επομένως, στη συνέχεια η αναζήτηση θα γίνει στο 2ο μισό



μέση

3η επανάληψη: μεταβλητή μέση παίρνει τιμή 3

Επειδή δεν ισχύει $KEY > A[μέση]$ και δεν ισχύει $KEY < A[μέση]$ η λογική μεταβλητή γίνεται ΑΛΗΘΗΣ οπότε η επανάληψη σταματά εδώ

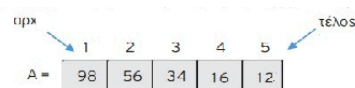
Επειδή ισχύει δεικτης = ΑΛΗΘΗΣ θα εμφανιστεί "Βρέθηκε στη θέση 3"

Περίπτωση 2^η Θεωρούμε ότι τα στοιχεία του πίνακα A είναι ταξινομημένα κατά φθίνουσα σειρά.

```

ΠΡΟΓΡΑΜΜΑ Διαδικής_Αναζήτηση2
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: A[5], KEY, τέλος, αρχ, μέση, i
  ΛΟΓΙΚΕΣ: δεικτης
ΑΡΧΗ
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
    ΔΙΑΒΑΣΕ A[i]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΡΑΨΕ 'Δώσε αριθμό για αναζήτηση στον πίνακα A.'
  ΔΙΑΒΑΣΕ KEY
  αρχ <- 1
  τέλος <- 5
  δεικτης <- ΨΕΥΔΗΣ
  ΟΣΟ αρχ <= τέλος ΚΑΙ δεικτης = ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ
    μέση <- (αρχ + τέλος) div 2
    ΑΝ μέση >= 1 ΚΑΙ μέση <= 5 ΤΟΤΕ
      ΑΝ KEY < A[μέση] ΤΟΤΕ
        αρχ <- μέση + 1
      ΑΛΛΙΩΣ_ΑΝ KEY > A[μέση] ΤΟΤΕ
        τέλος <- μέση - 1
      ΑΛΛΙΩΣ
        δεικτης <- ΑΛΗΘΗΣ
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΑΝ δεικτης = ΑΛΗΘΗΣ ΤΟΤΕ
    ΓΡΑΨΕ 'Βρέθηκε στη θέση ', μέση
  ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'Δε βρέθηκε'
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Διαδικής_Αναζήτηση2
    
```

Εστω ο πίνακας A[5], KEY= 14 και οι μεταβλητές αρχ, τέλος με την αρχικοποίηση

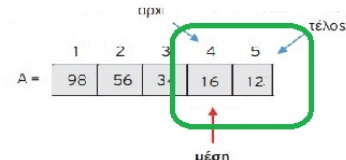


μέση

1η Επανάληψη: μεταβλητή μέση παίρνει τιμή 3

Επειδή ισχύει $KEY < A[μέση]$ η μεταβλητή αρχ γίνεται 4

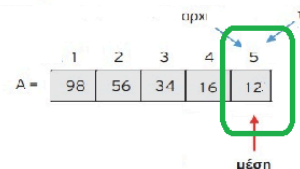
αλλάζει τιμή η μεταβλητή αρχ γιατί σε αυτή την περίπτωση έχουμε πίνακα με φθίνουσα ταξινόμηση οπότε την αναζήτηση την συνεχίζουμε στο 2ο μισό



μέση

2η Επανάληψη: μεταβλητή παίρνει τιμή 4

Επειδή ισχύει $KEY < A[4]$ η μεταβλητή αρχ γίνεται 5



μέση

3η Επανάληψη: μεταβλητή μέση παίρνει τιμή 5

Επειδή ισχύει $KEY > A[5]$ η μεταβλητή τέλος γίνεται 4

Δεν θα γίνει άλλη επανάληψη γιατί δεν ισχύει πλέον $αρχ <= τέλος$

Επειδή η λογική μεταβλητή δεικτης παρέμεινε ΨΕΥΔΗΣ θα εμφανιστεί "Δε βρέθηκε"