

ΑΝΑΖΗΤΗΣΗ ΣΕ ΣΤΑΤΙΚΕΣ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ (ΠΙΝΑΚΕΣ)

1.1

ΣΕΙΡΙΑΚΟΣ ΑΛΓΟΡΙΘΜΟΣ ΑΝΑΖΗΤΗΣΗΣ (ΓΕΝΙΚΕΣ ΠΑΡΑΤΗΡΗΣΕΙΣ)

ΠΑΡΑΤΗΡΗΣΕΙΣ ΣΕΙΡΙΑΚΟΥ ΑΛΓΟΡΙΘΜΟΥ

- Ο Σειριακός αλγόριθμος που υλοποιείται με την επαναληπτική δομή ΟΣΟ τερματίζεται όταν βρεθεί ο όρος αναζήτησης **ή** έχουν ελεγχθεί όλα τα στοιχεία του πίνακα στον οποίο αναζητούμε τον όρο
- Η λογική μεταβλητή done χρησιμοποιείται ως διακόπτης (flag) που ορίζει αν η αναζήτηση πρέπει να συνεχισθεί ή να ολοκληρωθεί(επειδή βρέθηκε το στοιχείο)
- Το αναζητούμενο στοιχείο μπορεί να είναι οποιουδήποτε τύπου
- Με βάση την παραλλαγή που παρουσιάσθηκε προηγουμένως, ο αλγόριθμος τερματίζει όταν βρεθεί το **1^ο στοιχείο του πίνακα που είναι ίσο με τον όρο αναζήτησης**. Αν ο πίνακας περιέχει και άλλα στοιχεία που είναι ίσα με τον όρο αναζήτησης, τότε αυτά δεν θα βρεθούν
- Αν θέλουμε το τελευταίο στοιχείο του πίνακα που είναι ίσο με τον όρο αναζήτησης, μπορούμε να ξεκινήσουμε την αναζήτηση από το τέλος του πίνακα(θέτοντας $i=N$ και όσο δεν βρίσκεται το στοιχείο να μειώνουμε το i κατά 1

ΠΑΡΑΤΗΡΗΣΕΙΣ ΣΕΙΡΙΑΚΟΥ ΑΛΓΟΡΙΘΜΟΥ

- Εφαρμόζεται σε όλους τους πίνακες και αποτελεί την πιο απλή μέθοδο αναζήτησης, αλλά και την λιγότερο αποδοτική/αποτελεσματική
- Χρησιμοποιούμε τον σειριακό αλγόριθμο όταν τα στοιχεία του πίνακα αναζήτησης δεν είναι ταξινομημένα, όταν δεν χρειάζεται συχνή υλοποίηση ή όταν το μέγεθος του πίνακα είναι μικρό (πχ 20 στοιχεία)
- Εφόσον τα στοιχεία ενός πίνακα είναι ταξινομημένα, τότε αποδοτικότερος αλγόριθμος είναι αυτός της **δυναμικής αναζήτησης**(θα εξετασθεί αργότερα)
- Με την τρέχουσα παραλλαγή του αλγορίθμου σειριακής αναζήτησης έχουμε την δυνατότητα να σταματάμε τον αλγόριθμο μόλις εντοπίσουμε το πρώτο στοιχείο του πίνακα που είναι ίσο με τον όρο αναζήτησης (ή το τελευταίο)
- Αν θέλουμε να εντοπίσουμε **όλα τα στοιχεία του πίνακα που είναι ίσα με τον όρο αναζήτησης, χρειάζεται να παρουσιάσουμε παραλλαγή του αλγόριθμου**

1.2

ΣΕΙΡΙΑΚΟΣ ΑΛΓΟΡΙΘΜΟΣ
ΑΝΑΖΗΤΗΣΗΣ
(ΑΝΑΖΗΤΗΣΗ ΤΟΥ
ΠΡΩΤΟΥ ΣΤΟΙΧΕΙΟΥ)

```

1 ΠΡΟΓΡΑΜΜΑ ΑΝΑΖΗΤΗΣΗ
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΧΑΡΑΚΤΗΡΕΣ: ΠΑΙΚΤΕΣ[5]
4   ΑΚΕΡΑΙΕΣ: i, pos
5   ΛΟΓΙΚΕΣ: done
6 ΑΡΧΗ
7   ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
8     ΓΡΑΨΕ 'Δώσε όνομα αθλητή'
9     ΔΙΑΒΑΣΕ ΠΑΙΚΤΕΣ[i]
10    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
11
12    done <- ΨΕΥΔΗΣ
13    pos <- 0
14    i <- 1
15
16    ΟΣΟ (done = ΨΕΥΔΗΣ) ΚΑΙ (i <= 5) ΕΠΑΝΑΛΑΒΕ
17      ΑΝ ΠΑΙΚΤΕΣ[i] = 'Nunn' ΤΟΤΕ
18        pos <- i
19        done <- ΑΛΗΘΗΣ
20      ΑΛΛΙΩΣ
21        i <- i + 1
22      ΤΕΛΟΣ_ΑΝ
23    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
24
25    ΑΝ done = ΑΛΗΘΗΣ ΤΟΤΕ
26      ΓΡΑΨΕ 'Ο Nunn βρέθηκε στη θέση ', pos
27    ΑΛΛΙΩΣ
28      ΓΡΑΨΕ 'Ο Nunn δεν βρέθηκε'
29    ΤΕΛΟΣ_ΑΝ

```

Δώσε όνομα αθλητή
grigonis
Δώσε όνομα αθλητή
flopenkof
Δώσε όνομα αθλητή
lessort
Δώσε όνομα αθλητή
Nunn
Ο Nunn βρέθηκε στη θέση 5

```

1 ΠΡΟΓΡΑΜΜΑ ΑΝΑΖΗΤΗΣΗ
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΧΑΡΑΚΤΗΡΕΣ: ΠΑΙΚΤΕΣ[5]
4   ΑΚΕΡΑΙΕΣ: i, pos
5   ΛΟΓΙΚΕΣ: done
6 ΑΡΧΗ
7   ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
8     ΓΡΑΨΕ 'Δώσε όνομα αθλητή'
9     ΔΙΑΒΑΣΕ ΠΑΙΚΤΕΣ[i]
10    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
11
12    done <- ΨΕΥΔΗΣ
13    pos <- 0
14    i <- 1
15
16    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
17      ΑΝ ΠΑΙΚΤΕΣ[i] = 'Nunn' ΤΟΤΕ
18        done <- ΑΛΗΘΗΣ
19        pos <- i
20      ΑΛΛΙΩΣ
21        i <- i + 1
22      ΤΕΛΟΣ_ΑΝ
23    ΜΕΧΡΙΣ_ΟΤΟΥ (done = ΑΛΗΘΗΣ) Η (i > 5)
24
25    ΑΝ done = ΑΛΗΘΗΣ ΤΟΤΕ
26      ΓΡΑΨΕ 'Ο Nunn βρέθηκε στη θέση ', pos
27    ΑΛΛΙΩΣ
28      ΓΡΑΨΕ 'Ο Nunn δεν βρέθηκε'
29    ΤΕΛΟΣ_ΑΝ

```

Και στις 2 περιπτώσεις, ο αλγόριθμος σταματάει μόλις εντοπίσει το πρώτο στοιχείο αναζήτησης. Μπορείτε να φανταστείτε τι τροποποιήσεις θα έπρεπε να κάνουμε για να σταματάει ο αλγόριθμος μόλις βρει το πρώτο στοιχείο από το τέλος του πίνακα;

1.3

ΣΕΙΡΙΑΚΟΣ ΑΛΓΟΡΙΘΜΟΣ
ΑΝΑΖΗΤΗΣΗΣ
(ΑΝΑΖΗΤΗΣΗ ΟΛΩΝ ΤΩΝ
ΣΤΟΙΧΕΙΩΝ)

```

1 ΠΡΟΓΡΑΜΜΑ ΑΝΑΖΗΤΗΣΗ
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΧΑΡΑΚΤΗΡΕΣ: ΠΑΙΚΤΕΣ[5]
4   ΑΚΕΡΑΙΕΣ: i
5 ΑΡΧΗ
6   ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
7     ΓΡΑΨΕ 'Δώσε όνομα αθλητή'
8     ΔΙΑΒΑΣΕ ΠΑΙΚΤΕΣ[i]
9   ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

10
11   i <- 1
12
13   ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
14     ΑΝ ΠΑΙΚΤΕΣ[i] = 'Nunn' ΤΟΤΕ
15       ΓΡΑΨΕ 'Βρέθηκε στην θέση ', i
16       i <- i + 1
17     ΤΕΛΟΣ_ΑΝ
18     i <- i + 1
19   ΜΕΧΡΙΣ_ΟΤΟΥ i > 5
20
21
22 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

ΜΕΧΡΙΣ ΟΤΟΥ

```

1 ΠΡΟΓΡΑΜΜΑ ΑΝΑΖΗΤΗΣΗ
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΧΑΡΑΚΤΗΡΕΣ: ΠΑΙΚΤΕΣ[5]
4   ΑΚΕΡΑΙΕΣ: i
5 ΑΡΧΗ
6   ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
7     ΓΡΑΨΕ 'Δώσε όνομα αθλητή'
8     ΔΙΑΒΑΣΕ ΠΑΙΚΤΕΣ[i]
9   ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

10
11   ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
12     ΑΝ ΠΑΙΚΤΕΣ[i] = 'Nunn' ΤΟΤΕ
13       ΓΡΑΨΕ 'Βρέθηκε στη θέση ', i
14     ΤΕΛΟΣ_ΑΝ
15   ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
16
17
18 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

ΓΙΑ

```

1 ΠΡΟΓΡΑΜΜΑ ΑΝΑΖΗΤΗΣΗ
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΧΑΡΑΚΤΗΡΕΣ: ΠΑΙΚΤΕΣ[5]
4   ΑΚΕΡΑΙΕΣ: i
5 ΑΡΧΗ
6   ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
7     ΓΡΑΨΕ 'Δώσε όνομα αθλητή'
8     ΔΙΑΒΑΣΕ ΠΑΙΚΤΕΣ[i]
9   ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

10
11   i <- 1
12   ΟΣΟ (i <= 5) ΕΠΑΝΑΛΑΒΕ
13     ΑΝ ΠΑΙΚΤΕΣ[i] = 'Nunn' ΤΟΤΕ
14       ΓΡΑΨΕ 'Βρέθηκε στη θέση ', i
15     ΤΕΛΟΣ_ΑΝ
16     i <- i + 1
17   ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
18
19
20 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

ΟΣΟ

Εφόσον μας ζητείται να εντοπίσουμε τις θέσεις όλων των στοιχείων του πίνακα που είναι ίσα με τον όρο αναζήτησης (χωρίς να αποθηκεύουμε τις θέσεις εντοπισμού)

1.4

ΣΕΙΡΙΑΚΟΣ ΑΛΓΟΡΙΘΜΟΣ

ΑΝΑΖΗΤΗΣΗΣ

(ΑΝΑΖΗΤΗΣΗ ΚΑΙ

ΑΠΟΘΗΚΕΥΣΗ ΘΕΣΕΩΝ ΟΛΩΝ

ΤΩΝ ΣΤΟΙΧΕΙΩΝ)

```

1 ΠΡΟΓΡΑΜΜΑ ΑΝΑΖΗΤΗΣΗ
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΧΑΡΑΚΤΗΡΕΣ: ΠΑΙΚΤΕΣ[5]
4   ΑΚΕΡΑΙΕΣ: i,j, δείκτης_θέσης, ΘΕΣΕΙΣ[5]
5 ΑΡΧΗ
6   ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
7     ΓΡΑΨΕ 'Δώσε όνομα αθλητή'
8     ΔΙΑΒΑΣΕ ΠΑΙΚΤΕΣ[i]
9     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
10
11   i <- 1
12   δείκτης_θέσης <- 0
13
14   ! Αποθήκευση όλων των θέσεων που βρέθηκε ο όρος αναζήτησης
15   ! σε νέο πίνακα
16   ΟΣΟ (i <= 5) ΕΠΑΝΑΛΑΒΕ
17     ΑΝ ΠΑΙΚΤΕΣ[i] = 'Nunn' ΤΟΤΕ
18       δείκτης_θέσης <- δείκτης_θέσης + 1
19       ΘΕΣΕΙΣ[δείκτης_θέσης] <- i
20     ΤΕΛΟΣ_ΑΝ
21     i <- i + 1
22   ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
23
24   ΑΝ δείκτης_θέσης <> 0 ΤΟΤΕ
25     ! Εμφάνιση όλων των θέσεων που βρέθηκε ο όρος αναζήτησης
26     ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ δείκτης_θέσης
27       ΓΡΑΨΕ ΘΕΣΕΙΣ[i]
28     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
29   ΑΛΛΙΩΣ
30     ΓΡΑΨΕ 'Δεν βρέθηκε το στοιχείο στον πίνακα'
31   ΤΕΛΟΣ_ΑΝ
32 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

Με ΟΣΟ

```

1 ΠΡΟΓΡΑΜΜΑ ΑΝΑΖΗΤΗΣΗ
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΧΑΡΑΚΤΗΡΕΣ: ΠΑΙΚΤΕΣ[5]
4   ΑΚΕΡΑΙΕΣ: i,j, δείκτης_θέσης, ΘΕΣΕΙΣ[5]
5 ΑΡΧΗ
6   ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
7     ΓΡΑΨΕ 'Δώσε όνομα αθλητή'
8     ΔΙΑΒΑΣΕ ΠΑΙΚΤΕΣ[i]
9     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
10
11   δείκτης_θέσης <- 0
12
13   ! Αποθήκευση όλων των θέσεων που βρέθηκε ο όρος αναζήτησης
14   ! σε νέο πίνακα
15   ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
16     ΑΝ ΠΑΙΚΤΕΣ[i] = 'Nunn' ΤΟΤΕ
17       δείκτης_θέσης <- δείκτης_θέσης + 1
18       ΘΕΣΕΙΣ[δείκτης_θέσης] <- i
19     ΤΕΛΟΣ_ΑΝ
20   ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
21
22
23   ΑΝ δείκτης_θέσης <> 0 ΤΟΤΕ
24     ! Εμφάνιση όλων των θέσεων που βρέθηκε ο όρος αναζήτησης
25     ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ δείκτης_θέσης
26       ΓΡΑΨΕ ΘΕΣΕΙΣ[i]
27     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
28   ΑΛΛΙΩΣ
29     ΓΡΑΨΕ 'Δεν βρέθηκε το στοιχείο στον πίνακα'
30   ΤΕΛΟΣ_ΑΝ
31
32 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

Με ΓΙΑ
(ΑΠΛΟΥΣΤΕΡΟ)

Εφόσον λοιπόν θέλουμε να αποθηκεύουμε ΟΛΕΣ ΤΙΣ ΘΕΣΕΙΣ που βρέθηκε ο όρος αναζήτησης, τις αποθηκεύουμε σε έναν νέο πίνακα (εδώ τον ονομάσαμε ΘΕΣΕΙΣ). Ο πίνακας αυτός έχει όσες θέσεις και ο αρχικός πίνακας, για να καλύψουμε την ακραία περίπτωση που όλα τα στοιχεία του αρχικού πίνακα είναι ίσα με τον όρο αναζήτησης