

Προγραμματισμός Ι (HY120)

Διάλεξη 10:
Ταξινόμηση Πίνακα – Αναζήτηση
σε Ταξινομημένο Πίνακα



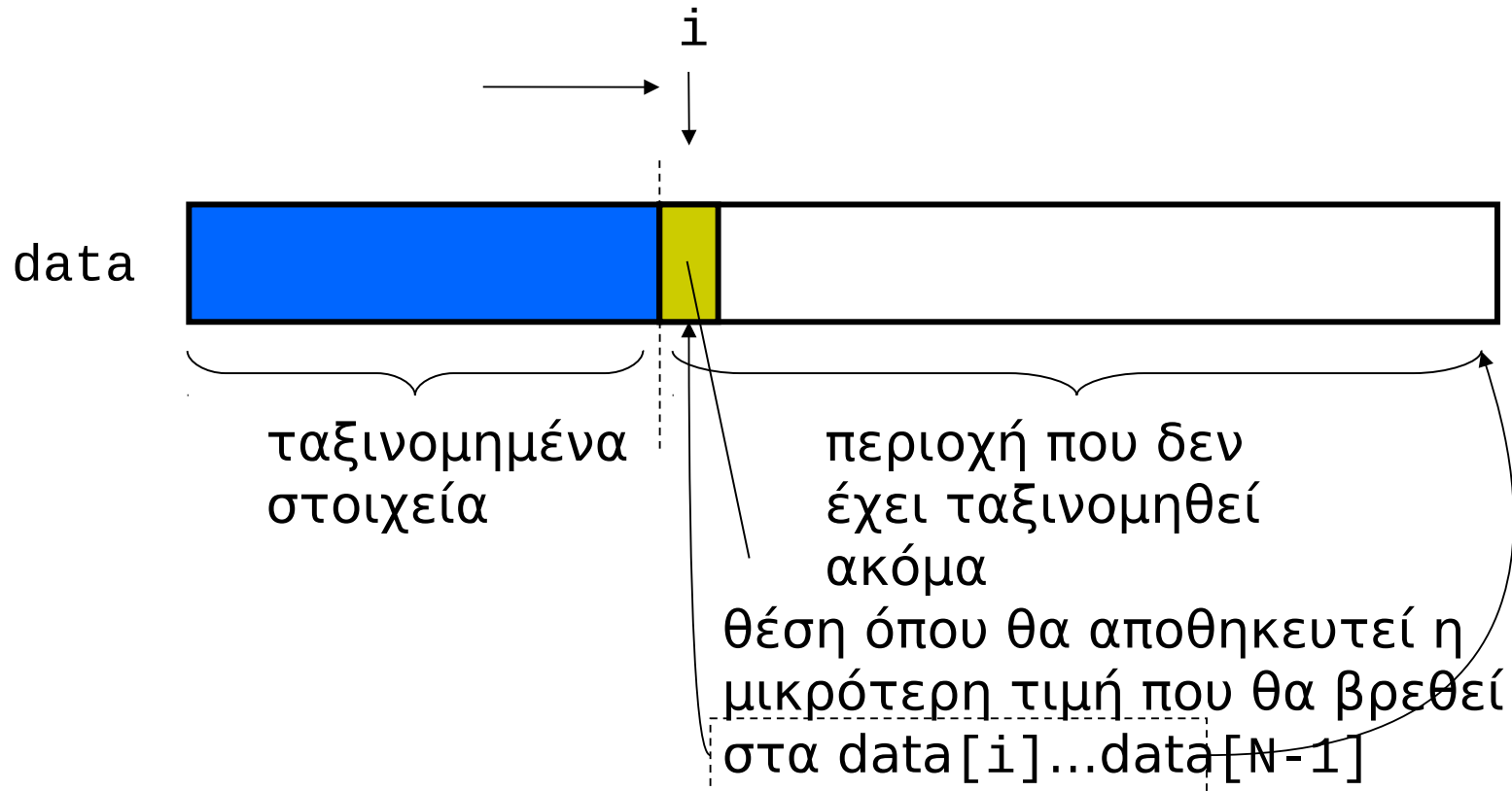


Πρόβλημα

- Δίνεται πίνακας t από N ακεραίους.
- Ζητούμενο: να ταξινομηθούν τα περιεχόμενα του πίνακα σε αύξουσα αριθμητική σειρά:

$$\forall i : 0 \leq i < N-1 \Rightarrow \text{data}[i] \leq \text{data}[i+1]$$

- Μια λύση (**ταξινόμηση επιλογής – selection sort**):
 - Αρχίζοντας από το 1ο στοιχείο και φτάνοντας μέχρι το N ο στοιχείο του πίνακα, βρίσκουμε το στοιχείο με την μικρότερη τιμή και ανταλλάσσουμε την τιμή του με την τιμή του 1ου στοιχείου.
 - Επαναλαμβάνουμε τη διαδικασία, αρχίζοντας από το 2ο στοιχείο, μετά αρχίζοντας από το 3ο στοιχείο κλπ μέχρι και το $N-1$ ο στοιχείο.



`/* ταξινόμηση πίνακα */`

```
int data[N];      /* πίνακας με N ακεραίους */
int i, j;          /* μετρητές βρόγχων */
int min;           /* θέση τοποθέτησης μικρότερου στοιχείου */
int tmp;           /* βοηθητική μεταβλητή για ανταλλαγή θέσης */
```

`... /* αρχικοποίηση πίνακα t με στοιχεία */`

```
for (i=0; i<N; i++) {
    min = i;
    for (j=i+1; j<N; j++) {
        if (data[j] < data[min]) {
            min = j;
        }
    }
    if (min != i) {
        tmp = data[i];
        data[i] = data[min];
        data[min] = tmp;
    }
}
```

*Ανταλλαγή τιμών 2
μεταβλητών (swap)*



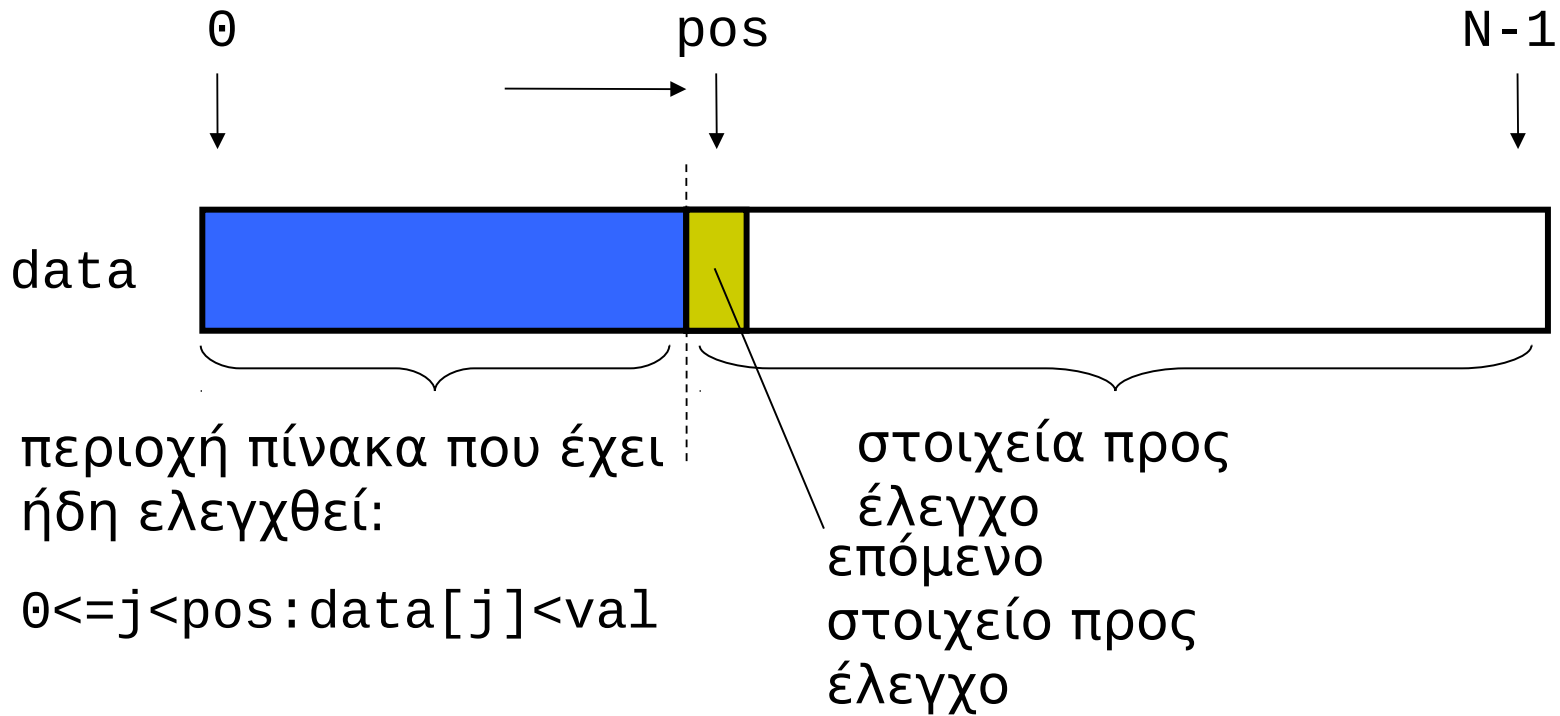
Πρόβλημα

- Δίνεται πίνακας t από N ακεραίους, τα περιεχόμενα του οποίου είναι **ταξινομημένα**.
- Ζητούμενο: Να βρεθεί η θέση όπου έχει αποθηκευθεί μια τιμή στον πίνακα (εφόσον υπάρχει)
 - Δηλαδή για μια τιμή v , να βρεθεί η τιμή i :
$$i \neq N \Rightarrow t[i] == v$$
$$i = N \Rightarrow \forall j : 0 \leq j < N : t[j] \neq v$$
- Εξετάζουμε δύο «κλασικές» λύσεις
 - ακολουθιακή αναζήτηση
 - δυαδική αναζήτηση (με διχοτόμηση)



Σειριακή αναζήτηση

- Αρχίζουμε από το πρώτο στοιχείο του πίνακα $\text{data}[\text{pos}]$, $\text{pos}=0$, που έχει (εξ' ορισμού) την μικρότερη τιμή και εκτελούμε σε βρόχο τις εντολές 2 έως 5.
 - Αν $\text{pos}==N$, δηλαδή ο πίνακας δεν έχει στοιχείο $\text{data}[\text{pos}]$, τότε τελειώσαμε (δεν υπάρχει στοιχείο με τιμή val).
 - Διαφορετικά, αν $\text{data}[\text{pos}]==\text{val}$, τότε τελειώσαμε (αφού βρήκαμε το στοιχείο με τιμή val).
 - Διαφορετικά, αν $\text{data}[\text{pos}]>\text{val}$, τότε πάλι τελειώσαμε (είναι σίγουρο ότι δεν υπάρχει στοιχείο με τιμή val).
 - Διαφορετικά, (ισχύει $\text{data}[\text{pos}]<\text{val}$), τότε επαναλαμβάνουμε την διαδικασία από το 2 για το επόμενο στοιχείο ($\text{pos}=\text{pos}+1$).
- Χρειάζονται κατά μέσο όρο «τάξη μεγέθους» $N/2$ βήματα σύγκρισης.



/* σειριακή αναζήτηση σε ταξινομημένο πίνακα */

int data[N]; /* πίνακας με N ακεραίους */

int pos; /* τρέχουσα θέση */

int val; /* τιμή στοιχείου που αναζητούμε */

... /* αρχικοποίηση & ταξινόμηση πίνακα data */

for (pos=0; pos<N; pos++) {

if (data[pos] >= val) {

break;

}

}

if (pos == N) {

printf("not found\n");

}

else {

if (data[pos] != val) {

printf("not found\n");

}

else {

printf("found at position %d\n", pos);

}

}





```
/* σειριακή αναζήτηση σε ταξινομημένο πίνακα */

int data[N]; /* πίνακας με N ακεραίους */
int pos;     /* μεταβλητή βρόγχου */
int val;     /* τιμή στοιχείου που αναζητούμε */

... /* αρχικοποίηση & ταξινόμηση πίνακα t */

for (pos=0; (pos<N) && !(data[pos]<val); pos++) {}

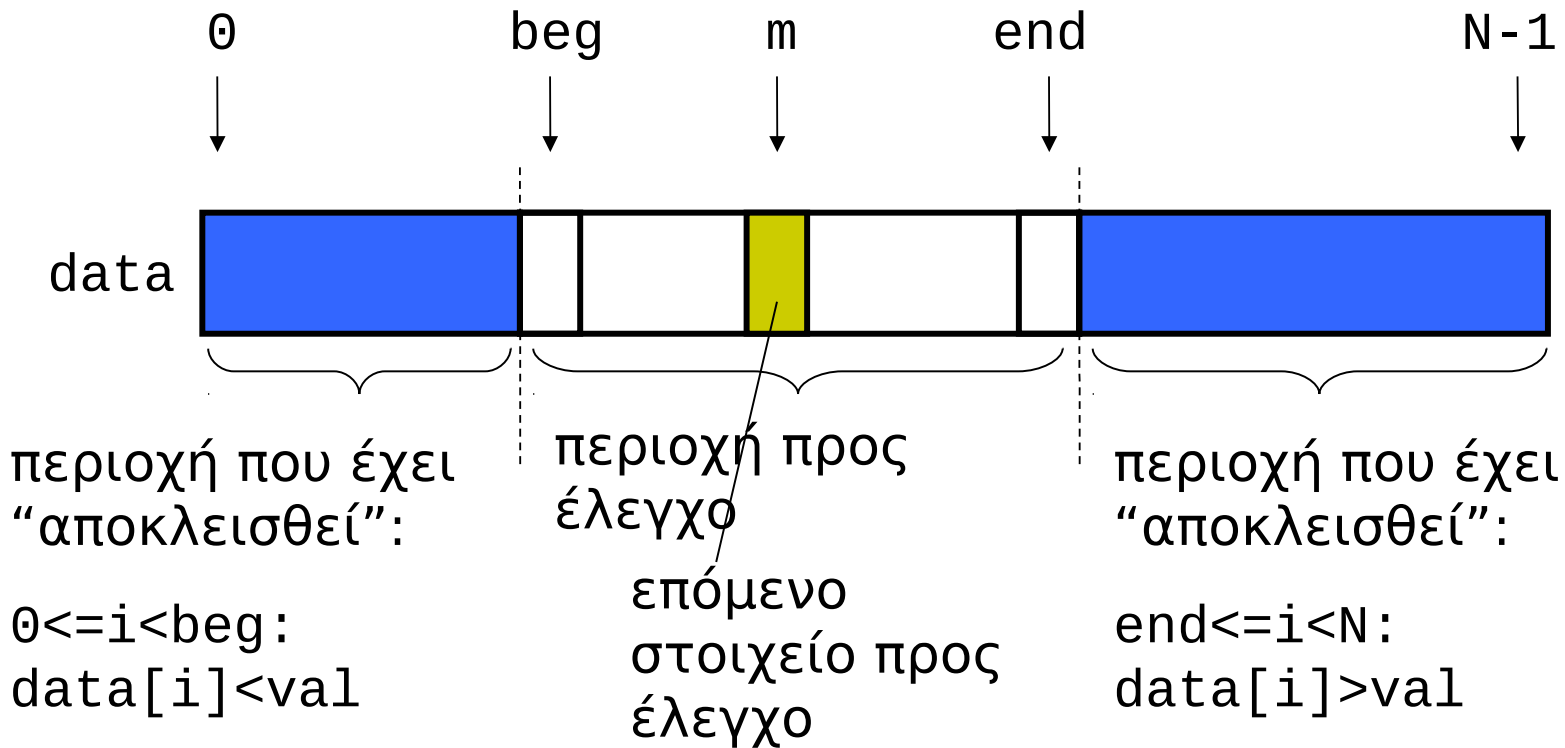
if (pos == N) {
    printf("not found\n");
}
else
    if (data[pos] != val) {
        printf("not found\n");
    }
    else {
        printf("found at position %d\n", pos);
    }
}
```

η σύμβαση εκτέλεσης του &&
εγγυάται ότι **δεν** θα επιχειρηθεί
πρόσβαση t[i] όταν το i φτάσει
το όριο N του πίνακα



Δυαδική αναζήτηση

- Θέτουμε τα όρια αναζήτησης $beg=0$ και $end=N-1$.
- Αρχίζουμε από το μεσαίο στοιχείο του πίνακα $t[m]$, όπου $m = (beg+end)/2$.
- Αν $t[m] == v$, τελειώσαμε.
- Αν $t[m] < v$, συνεχίζουμε με τον (μισό) υποπίνακα που έχει στοιχεία με **μεγαλύτερες** τιμές του $t[m]$ δηλαδή θέτουμε $beg = m+1$ (και πίσω στο 2).
- Αλλιώς ($t[m] > v$), συνεχίζουμε με τον (μισό) υποπίνακα που έχει στοιχεία με **μικρότερες** τιμές του $t[m]$ δηλαδή θέτουμε $end = m-1$ (και πίσω στο 2).
- Αν $beg > end$ τελειώσαμε (η τιμή v δεν υπάρχει).
- Χρειάζονται «τάξη μεγέθους» $\log_2 N$ βήματα.





```
/* δυαδική αναζήτηση σε ταξινομημένο πίνακα */  
int data[N]; /* πίνακας με N ακεραίους */  
int beg, end; /* όρια αναζήτησης [beg, end) */  
int m; /* θέση μεσαίου στοιχείου */  
int val; /* τιμή στοιχείου που αναζητούμε */
```

```
... /* αρχικοποίηση & ταξινόμηση πίνακα t */  
beg = 0;  
end = N-1;  
while (beg <= end) {  
    m = (beg+end)/2;  
    if (data[m] == val) {  
        break;  
    }  
    else {  
        if (data[m] < val) {  
            beg = m+1;  
        }  
        /* (data[m] > val) */  
        else {  
            end = m-1;  
        }  
    }  
}
```

```
if (beg > end) {  
    printf("not found\n");  
}  
else {  
    printf("found at position %d\n", m);  
}
```



Σύγκριση

# στοιχείων πίνακα <i>N</i>	μ.ο βημάτων σειριακής αναζ. <i>N/2</i>	μέγιστα βήματα δυαδικής αναζ. <i>log N</i>
100	50	7
1.000	500	10
1.000.000	500.000	20
1.000.000.000	500.000.000	30

- Η διαφορά γίνεται **τεράστια** για μεγάλες τιμές του N .