



# Αλγόριθμοι και Δομές Δεδομένων(Θ)

## Ενότητα 6: ΤΑΞΙΝΟΜΗΣΗ

Ευάγγελος Γ. Ούτσιος

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕ



# Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Κεντρικής Μακεδονίας» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



# ΤΑΞΙΝΟΜΗΣΗ

Ταξινόμηση είναι η διαδικασία της τοποθέτησης ενός συνόλου στοιχείων σε μία ιδιαίτερη σειρά.

Η σειρά αυτή είναι συνήθως αύξουσα (*ascending*) ή φθίνουσα (*descending*).

Σκοπός της ταξινόμησης είναι η διευκόλυνση της αναζήτησης στοιχείων του ταξινομημένου συνόλου.

# ΤΑΞΙΝΟΜΗΣΗ

## ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΑΠΕΥΘΕΙΑΣ ΕΠΙΛΟΓΗ (Straight Selection)

Η μέθοδος έχει ως εξής:

- Επιλέγουμε το μικρότερο στοιχείο
- Το ανταλλάσσουμε με το πρώτο στοιχείο
- Επαναλαμβάνουμε για τα υπόλοιπα στοιχεία, μέχρι να μείνει ένα

Π.χ.

<b>44</b>	55	12	42	94	18	<b>06</b>	67
06	<b>55</b>	<b>12</b>	42	94	18	44	67
06	12	<b>55</b>	42	94	<b>18</b>	44	67
06	12	18	<b>42</b>	94	55	44	67
06	12	18	42	<b>94</b>	55	<b>44</b>	67
06	12	18	42	44	<b>55</b>	94	67
06	12	18	42	44	55	<b>94</b>	<b>67</b>
06	12	18	42	44	55	67	94

# ΤΑΞΙΝΟΜΗΣΗ

Ο αλγόριθμος σε C:

```
for (i=0; i<N-1; i++)
{
    k = i;
    min = p[i];
    for (j = i+1; j<N; j++)
        {
            if (p[j] < min)
                {
                    k = j;
                    min = p[j];
                }
        }
    p[k] = p[i] ;
    p[i] = min;
}
```

# ΤΑΞΙΝΟΜΗΣΗ

## ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΑΠΕΥΘΕΙΑΣ ΕΙΣΑΓΩΓΗ (Straight Insertion)

Η μέθοδος είναι πολύ δημοφιλής στους χαρτοπαίχτες και έχει ως εξής:

- Κάθε στοιχείο, ξεκινώντας από το δεύτερο, τοποθετείται στη σωστή του θέση μετακινώντας, αν χρειαστεί, τα στοιχεία δεξιά του κατά μία θέση.

Π.χ.

44	<b>55</b>	12	42	94	18	06	67
44	55	<b>12</b>	42	94	18	06	67
12	44	55	<b>42</b>	94	18	06	67
12	42	44	55	<b>94</b>	48	06	67
12	42	44	55	94	<b>18</b>	06	67
12	18	42	44	55	94	<b>06</b>	67
06	12	18	42	44	55	94	<b>67</b>
06	12	18	42	44	55	67	94

# ΤΑΞΙΝΟΜΗΣΗ

Ο αλγόριθμος σε C:

```
for (i=2; i<=N; i++)  
{  
    x = p[i];  
    p[0] = x;  
    j = i-1;  
    while (x < p[j])  
        {  
            p[j+1] = p[j] ;  
            j = j-1 ;  
        }  
    p[j+1] = x ;  
}
```



# ΤΑΞΙΝΟΜΗΣΗ

## ΤΑΞΙΝΟΜΗΣΗ ΦΥΣΑΛΙΔΑΣ (Bubble Sort)

- Ο αλγόριθμος βασίζεται στην αρχή της σύγκρισης και ανταλλαγής ζευγών από γειτονικά στοιχεία, μέχρις ότου ταξινομηθούν όλα τα στοιχεία. Κάθε φορά μετακινείται το μικρότερο στοιχείο της ακολουθίας προς το αριστερό άκρο.

Π.χ.

44	55	12	42	94	18	06	67
06	44	55	12	42	94	18	67
06	12	44	55	18	42	94	67
06	12	18	44	55	42	67	94
06	12	18	42	44	55	67	94
06	12	18	42	44	55	67	94
06	12	18	42	44	55	67	94
06	12	18	42	44	55	67	94

# ΤΑΞΙΝΟΜΗΣΗ

Ο αλγόριθμος σε C:

```
for (i=1; i<N; i++)  
    for (j=N-1; j>=i; j--)  
        if (p[j-1] > p[j])  
            {  
                temp = p[j-1];  
                p[j-1] = p[j] ;  
                p[j] = temp ;  
            }
```

# ΤΑΞΙΝΟΜΗΣΗ

## ΓΡΗΓΟΡΗ ΤΑΞΙΝΟΜΗΣΗ (Quick Sort)

- Ο αλγόριθμος βασίζεται στην αρχή της αντιμετάθεσης και είναι η καλύτερη γνωστή μέθοδος ταξινόμησης για τυχαία στοιχεία. Η γρήγορη ταξινόμηση στηρίζεται στην παρατήρηση ότι είναι προτιμότερο οι αντιμεταθέσεις να γίνονται μεταξύ απομακρυσμένων στοιχείων.
- Στην αρχή λαμβάνεται το μεσαίο στοιχείο του πίνακα και μετακινείται στη θέση όπου τελικά θα αποθηκευτεί στο ταξινομημένο διάνυσμα. Μετά τον προσδιορισμό της τελικής αυτής θέσης, γίνεται αναδιάταξη των υπόλοιπων στοιχείων έτσι ώστε να μην υπάρχει κανένα μικρότερο στοιχείο προς τα αριστερά του και κανένα μεγαλύτερο στοιχείο προς τα δεξιά του.

# ΤΑΞΙΝΟΜΗΣΗ

- Έτσι, το μεσαίο αυτό στοιχείο παίζει το ρόλο του άξονα (pivot) και ο πίνακας έχει διαμεριστεί κατά τέτοιο τρόπο ώστε το αρχικό πρόβλημα έχει αναχθεί σε δύο απλούστερα προβλήματα, στην ανεξάρτητη δηλαδή ταξινόμηση των δύο υποπίνακων.
- Μετά τον διαμερισμό του πίνακα, η ίδια διαδικασία εφαρμόζεται στους δύο υποπίνακες, ύστερα στους υποπίνακες των υποπίνακων, κοκ., μέχρις ότου ο κάθε υποπίνακας να αποτελείται από ένα μόνο στοιχείο. Στο σημείο αυτό ο αρχικός πίνακας έχει ταξινομηθεί.

# ΤΑΞΙΝΟΜΗΣΗ

Ο αλγόριθμος σε C:

```
void quicksort(int left, int right, int p[])
```

```
{  
    int i, j, mid, x, temp;  
  
    if (left < right)  
    {  
        i = left;  
        j = right;  
        mid = (left+right)/2;  
        x = p[mid];  
        while (i < j)  
        {  
            while (p[i] < x)  
                i++;  
            while (p[j] > x)  
                j--;  
        }  
    }  
}
```

```
    if (i < j)  
    {  
        if (p[i] == p[j])  
        {  
            if (i < mid)  
                i++;  
            if (j > mid)  
                j--;  
        }  
        else  
        {  
            temp = p[i];  
            p[i] = p[j];  
            p[j] = temp;  
        }  
    }  
    quicksort(left, j-1, p);  
    quicksort(j+1, right, p);  
}
```