# ΑΕΠΠ - ΚΕΦΑΛΑΙΟ 3ο

## ΘΕΩΡΙΑ- ΤΑΞΙΝΟΜΗΣΗ

### 1. Ορισμός

* Η **τακτοποίηση** των κόμβων μίας δομής με μία ιδιαίτερη **σειρά** είναι μία πολύ σημαντική λειτουργία που ονομάζεται **ταξινόμηση** (sorting) ή διάταξη (ordering).

**Ορισμός.** Δοθέντων των στοιχείων a1,a2,...,an η ταξινόμηση συνίσταται στη μετάθεση (permutation) της θέσης των στοιχείων, ώστε να τοποθετηθούν σε μία σειρά ak1,ak2,...,akn έτσι ώστε, δοθείσης μίας συνάρτησης διάταξης (ordering function), f, να ισχύει: f(ak1) ≤ f(ak2) ≤ ... ≤ f(akn)

* Aξίζει να σημειωθεί ότι η προηγούμενη συνάρτηση διάταξης μπορεί νατροποποιηθεί, ώστε να καλύπτει και την περίπτωση που η ταξινόμηση γίνεται με φθίνουσα τάξη (descending sequence) μεγέθους.

### 3. Ταξινόμηση ευθείας ανταλλάγής

* H μέθοδος της **ταξινόμησης ευθείας ανταλλαγής** (straight exchange sort) βασίζεται στην αρχή της **σύγκρισης και ανταλλαγής ζευγών γειτονικών στοιχείων**, μέχρις ότου διαταχθούν όλα τα στοιχεία.
* Σύμφωνα με τη μέθοδο αυτή κάθε φορά γίνονται **διαδοχικές προσπελάσεις** στον πίνακα απο το τελευταίο στοιχείο και πραγματοποιούνται διαδοχικές συγκρίσεις συνεχόμενων θέσεων.
* Αν το στοιχείο της μεγαλύτερης θέσης είναι μικρότερο απο το στοιχείο της προηγούμενης της, τότε στοιχεία πρέπει να αντιμετατεθούν.( περίπτωση αύξουσας ταξινόμησης).

#### Παράδειγμα: Έστω ο παρακάτω πίνακας table:

|  |
| --- |
| 5 |
| 4 |
| 3 |
| 2 |

|  |  |
| --- | --- |
| **1η Προσπελαση** | **Τμήμα Αλγορίθμου** |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  | | --- | | 5 | | 4 | | **3** | | **2** | | |  | | --- | | 5 | | **4** | | **2** | | 3 | | |  | | --- | | **5** | | **2** | | 4 | | 3 | | |  | | --- | | **2** | | 5 | | 3 | | 4 | | | **1η** σύγκριση:  4ου-3ου στοιχείου | **2η** σύγκριση:  3ου-2ου στοιχείου | **3η** σύγκριση:  2ου-1ου στοιχείου | Το **μικρότερο**στοιχείο τοποθετείται στην πρώτη θέση | |  |  |  |  | | Για j από 4 μέχρι **2** με\_βήμα -1  Αν table[j]<table[j-1] τότε  temp← table[j]  table[j]← table[j-1]  table[j-1]← temp  Τέλος\_αν  Τέλος\_επανάληψης |

|  |  |
| --- | --- |
| **2η Προσπελαση** | **Τμήμα Αλγορίθμου** |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  | | --- | | 2 | | 5 | | **3** | | **4** | | |  | | --- | | 2 | | **5** | | **3** | | 4 | | |  | | --- | | **2** | | **3** | | 5 | | 4 | | | **1η** σύγκριση:  4ου-3ου στοιχείου | **2η** σύγκριση:  3ου-2ου στοιχείου | Τα**2 μικρότερα** στοιχεία τοποθετούνται στις 2 πρώτες θέσεις | |  |  |  | | Για j από 4 μέχρι **3** με\_βήμα -1  Αν table[j]<teble[j-1] τότε  temp← table[j]  table[j]← table[j-1]  table[j-1]← temp  Τέλος\_αν  Τέλος\_επανάληψης |

|  |  |
| --- | --- |
| **3η Προσπελαση** | **Τμήμα Αλγορίθμου** |
| |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  | | --- | | 2 | | 3 | | **5** | | **4** | | |  | | --- | | **2** | | **3** | | 4 | | 5 | | | **1η** σύγκριση:  4ου-3ου στοιχείου | Τα **3 μικρότερα** στοιχεία τοποθετούνται στις 3 πρώτες θέσεις | |  |  | | Για j από 4 μέχρι **4** με\_βήμα -1  Αν table[j]<table[j-1] τότε  temp← table[j]  table[j]← table[j-1]  table[j-1]← temp  Τέλος\_αν  Τέλος\_επανάληψης |

* Σε κάθε προσπέλαση πραγματοποιείται μια σύγκριση λιγότερη
* Τα τμήματα των αλγόριθμων που υλοποιούν την κάθε προσπέλαση διαφοροποιούνται μόνο σχετικά με την τελική τιμή του j (ποιό στοιχείο του πίνακα θα συγκριθεί τελευταίο με το προηγούμενο του).
* Αν χρησιμοποιήσουμε μια μεταβλητή **i** που θα παίρνει διαδοχικά **τις τελικές του j σε κάθε προσπέλαση**, η ταξινόμηση του παραπάνω πίνακα μπορεί να γραφτεί σύμφωνα με το παρακάτω αλγόριθμο:

Αλγόριθμος Φυσαλιδα

Δεδομένα //table//

Για **i** από 2 μέχρι 4

Για j από 4 μέχρι i με\_βήμα -1

Αν table[j]<table[j-1] τότε

temp← table[j]

table[j]← table[j-1]

table[j-1]← temp

Τέλος\_αν

Τέλος\_επανάληψης

Αποτελέσματα //table//

Τέλος Φυσαλιδα

* Στην γενικότερη περίπτωση **πίνακα table nστοιχείων** ο αλγόριθμος παίρνει την παρακάτω μορφή:

Αλγόριθμος Φυσαλιδα

Δεδομένα //table, n//

Για **i** από 2 μέχρι **n**

Για j από **n** μέχρι i με\_βήμα -1

Αν table[j]<table[j-1] τότε

temp← table[j]

αντιμετάθεσε table[j],table[j-1]

table[j]← table[j-1]

table[j-1]← temp

Τέλος\_αν

Τέλος\_επανάληψης

Αποτελέσματα //table//

Τέλος Φυσαλιδα

### ΠΑΡΑΤΗΡΗΣΕΙΣ

* O αλγόριθμος της ευθείας ανταλλαγής μπορεί εφαρμοστεί και σε πίνακες χαρακτήρων
* Για **φθίνουσα** ταξινόμηση το μόνο που αλλάξει είναι η **φορά της ανίσωσης** στη δομή επιλογής (table[j] **>**table[j-1]).
* Ο πίνακας table είναι μονοδιάστατος. Δεν μπορεί να χρησιμοποιηθεί για δισδιάστατους ως προς τις δυο τους διαστάσεις.
* Κατά την ταξινόμηση γίνονται **n-1**προσπελάσεις.
* Στην τελευταία προσπέλαση γίνεται μια μόνο ταξινόμηση.
* Για την ταξινόμηση δεδομένων έχουν εκπονηθεί πάρα πολλοί αλγόριθμοι. Άλλοι σχετικά απλοί αλγόριθμοι είναι
  + η **ταξινόμηση με επιλογή** και
  + η ταξινόμηση **με παρεμβολή.**
  + Ο πιο γρήγορος αλγόριθμος ταξινόμησης είναι η “**γρήγορη ταξινόμηση” (quicksort).**
* Η ταξινόμηση **φυσσαλίδας** είναι ο **πιο απλός** και ταυτόχρονα ο **πιο αργός** αλγόριθμος ταξινόμησης.