

## ΑΣΚΗΣΕΙΣ

### Αλγόριθμοι Ταξινόμησης (*Sorting Algorithms*)

#### **ΑΣΚΗΣΗ-1<sup>η</sup>**

Να γραφεί το πρόγραμμα Java που ταξινομεί ένα πίνα ακεραίων με τον αλγόριθμο της **ταξινόμησης φυσαλίδας** (*bubble sort*). Ο πίνακας θα εμφανίζεται πριν και μετά την ταξινόμηση. Γράψτε την στατική μέθοδο **BubbleSort(int[] ar)** που θα δέχεται σαν παράμετρο τον αταξινομητο πίνακα των ακεραίων και θα υλοποιεί την ταξινόμηση. Χρησιμοποιήστε μια boolean μεταβλητή-flag που θα βοηθήσει στο να τερματιστεί η ταξινόμηση αν δεν έχει συμβεί κάποια ανταλλαγή τιμών (επομένως ο πίνακας είναι ταξινομημένος).

#### **ΑΣΚΗΣΗ-2<sup>η</sup>**

Να γραφεί το πρόγραμμα Java που ταξινομεί ένα πίνα ακεραίων με τον αλγόριθμο της **ταξινόμησης με εισαγωγή** (*insertion sort*). Ο πίνακας θα εμφανίζεται πριν και μετά την ταξινόμηση. Γράψτε την στατική μέθοδο **InsertionSort(int[] ar)** που θα δέχεται σαν παράμετρο τον αταξινομητο πίνακα των ακεραίων και θα υλοποιεί την ταξινόμηση. Υπάρχουν πολλές παραλλαγές του αλγορίθμου. Να υλοποιήσετε την Insertion Sort και με 2<sup>ο</sup> αλγόριθμο.

#### **ΑΣΚΗΣΗ-3<sup>η</sup>**

Να γραφεί το πρόγραμμα Java που ταξινομεί ένα πίνα ακεραίων με τον αλγόριθμο της **ταξινόμησης με επιλογή** (*selection sort*). Ο πίνακας θα εμφανίζεται πριν και μετά την ταξινόμηση. Γράψτε την στατική μέθοδο **SelectionSort(int[] ar)** που θα δέχεται σαν παράμετρο τον αταξινομητο πίνακα των ακεραίων και θα υλοποιεί την ταξινόμηση. Υπάρχουν πολλές παραλλαγές του αλγορίθμου. Προσοχή ο αλγόριθμος πρέπει να 'τρέχει - δουλεύει' σωστά για μονό και ζυγό πλήθος στοιχείων του πίνακα.

#### **ΑΣΚΗΣΗ-4<sup>η</sup> (παραλλαγή της 3<sup>ης</sup>)**

Να γραφεί το πρόγραμμα Java που ταξινομεί ένα πίνα ακεραίων με τον αλγόριθμο της **αναδρομικής γρήγορης ταξινόμησης** (*quick sort*). Γράψτε την στατική μέθοδο **QuickSort(int[] ar, int first, int last)** που θα καλείται αναδρομικά και θα υλοποιεί την ταξινόμηση. Επίσης χρησιμοποιήστε την στατική μέθοδο: **partition(int[] ar, int first, int last)**, που θα εκτελεί το partitioning (συνεχή διαχωρισμό του πίνακα με χρήση της τιμής pivot).

**Προσοχή !!!** Υπάρχουν πολλές παραλλαγές του αλγορίθμου. Ο αλγόριθμος πρέπει να 'τρέχει - δουλεύει' σωστά για μονό και ζυγό πλήθος στοιχείων του πίνακα. Η προτεινόμενη λύση είναι από το βιβλίο του Liang. Στις λύσεις προτείνεται και 2<sup>η</sup> παραλλαγή (με χρήση αντικειμένου).

### **ΑΣΚΗΣΗ-5<sup>η</sup>**

Να γραφεί το πρόγραμμα Java που ταξινομεί ένα πίνα ακεραίων με τον αλγόριθμο της **αναδρομικής ταξινόμησης με συγχώνευση** (*merge sort*). Γράψτε την στατική μέθοδο:

**MergeSort(int[] ar)**, που θα καλείται αναδρομικά και και την στατική μέθοδο:

**merge(int[] list1, int[] list2, int[] temp)**, που θα υλοποιεί την συνένωση των δύο πινάκων με την βοήθεια του temp[].

### **ΑΛΥΤΕΣ ΑΣΚΗΣΕΙΣ**

#### **ΑΣΚΗΣΗ-1<sup>η</sup>** (Παραλλαγή της άσκησης στο προηγούμενο μάθημα)

Να γίνει το πρόγραμμα αναζήτησης και εμφάνισης των στοιχείων ενός υπάλληλου σε ένα πίνακα **N** - αντικειμένων τύπου Employee. Ορίστε την κλάση **Employee** με private πεδία: (1) Όνομα, String, (2) Επώνυμο, String, (3) Κωδικός Υπαλ, int και (4) Μισθός, double.

Το πρόγραμμα θα εκτελεί (1) σειριακή αναζήτηση ως προς το Επώνυμο και (2) δυαδική αναζήτηση ως προς τον Κωδικό Υπαλ. (σε ταξινομημένο ως προς τον Κωδικό-πίνακα). Οι εργασίες θα υλοποιούνται από ένα **μενού επιλογών**.

Οι εργασίες θα καλούνται κατάλληλα, δηλ. δεν θα μπορείς να κάνεις δυαδική αναζήτηση, αν δεν έχει **εκτελεστεί πρώτα η ταξινόμηση** ως προς τον κωδικό (όποια ταξινόμηση θέλετε εσείς). Θα εμφανίζονται τα κατάλληλα μηνύματα που θα καθοδηγούν τις εργασίες.

**Για την σειριακή αναζήτηση:** Το πρόγραμμα θα ζητά το Επώνυμο του υπάλληλου που αναζητάμε. Αν το βρει θα εμφανίσει όλα τα στοιχεία του, διαφορετικά το μήνυμα "Den yparxei o Ypallilos".

**Για την δυαδική αναζήτηση:** Το πρόγραμμα θα ζητά τον κωδικό του και θα υλοποιεί τα ίδια με την σειριακή αναζήτηση (εμφάνιση στοιχείων).