

Project 2:

1. Στην ενότητα «2.5 Sorting Applications (<http://algs4.cs.princeton.edu/25applications/>)» δίνεται η κλάση Student.java (δείτε Web exercises άσκηση 3) καθώς και η κλάση Grade.java (δείτε Web exercises άσκηση 2) καθώς και η κλάση Transaction.java (στην αρχή της ενότητας 2.5). Λαμβάνοντας υπόψη τις παραπάνω κλάσεις να τροποποιήσετε την κλάση Student ώστε για κάθε φοιτητή εκτός από το όνομα (name), και το τμήμα (section), να προσθέσετε πεδία για: την ημερομηνία εγγραφής στο τμήμα (τύπου Date), το ύψος του (τύπου double) και το χαρακτηρισμό της επίδοσης (τύπου Grade) και υλοποιήστε επιπλέον comparators για ταξινόμηση με βάση την ημερομηνία εγγραφής, με βάση το ύψος, με βάση το χαρακτηρισμό της επίδοσης και comparator για φθίνουσα ταξινόμηση ως προς το όνομα (name).
Υλοποιήστε κλάση πελάτη (p2e1.java) όπου θα ταξινομείτε τους φοιτητές με βάση το όνομα, το όνομα σε φθίνουσα διάταξη, το τμήμα, την ημερομηνία, το ύψος, το χαρακτηρισμό της επίδοσης, και με βάση το δικό σας όνομα. Για τις παραπάνω ταξινομήσεις καλέστε εναλλάξ την sort της κλάσης Arrays και την κατάλληλη merge που δίνεται στην ενότητα 2.2 (MergeX.java). Προσθέστε φοιτητή με τα δικά σας στοιχεία, και βέβαια δημιουργήστε αντικείμενα τη κλάσης Student με όλα τα παραπάνω πεδία. Η δημιουργία των αντικειμένων τύπου Student μπορεί να γίνει όπως στην main της test class της Student.java. Για κάθε φοιτητή θα εμφανίζετε τα στοιχεία του ως εξής:
όνομα: τμήμα: ύψος: χαρακτηρισμός επίδοσης: ημερομηνία
Παραδοτέο: p2e1.zip θα περιλαμβάνει όλα τα .java & class αρχεία.
2. Δίνεται ένα maximum binary heap που περιέχει n αριθμούς (στη ρίζα αποθηκεύεται ο μεγαλύτερος). Δίνεται και ένας θετικός ακέραιος $k < n$ και ο αριθμός x. Πρέπει να προσδιορίσετε αν το k-οστό μεγαλύτερο στοιχείο του max binary heap είναι μεγαλύτερο ή όχι από το x. Οι τιμές για το k και x θα δίνονται ως παράμετροι της main. Θα εμφανίζετε το maximum binary heap καθώς και το αν το k-οστό μεγαλύτερο στοιχείο του maximum binary heap είναι μεγαλύτερο ή όχι από το x. Η εμφάνιση θα είναι ως εξής:
Αν για πχ ως είσοδο δοθούν τα εξής δεδομένα: 19 31 27 33 10 35 26 42 14 44 και ως α) $k=5$ $x=27$ ή β) $k=5$ $x=35$ θα εμφανίσει αντίστοιχα:

α) The max binary heap is: 44 42 35 33 31 27 26 19 14 10 Yes. The 5th largest item is greater than 27	β) The max binary heap is: 44 42 35 33 31 27 26 19 14 10 No. The 5th largest item isn't greater than 35
--	--

Παραδοτέο: p2e2.zip θα περιλαμβάνει όλα τα .java & class αρχεία και δεδομένα I/O(*).

3. (<https://algs4.cs.princeton.edu/25applications/> exercise 20). Min and max. Given an array of N elements, find the min and max using as few compares as possible. Brute force: find the max (N-1 compares), then find the min of the remaining elements (N-2 compares).
Solution 1. Divide and conquer: find min and max in each half ($2T(N/2)$ compares), return min of 2 and max of 2 (2 compares). $T(1) = 0$, $T(2) = 1$, $T(N) = 2T(N/2) + 2$. Recurrence solution: $T(N) = \text{ceil}(3N/2) - 2$.
Solution 2. Divide the elements into pairs and compare two elements in each pair. Put the smallest elements in A and the largest in B. If n is odd, put element n in both A and B. This requires $\text{floor}(n/2)$ comparisons. Now directly compute the minimum in A ($\text{ceil}(n/2) - 1$ comparisons) and the maximum in B ($\text{ceil}(N/2) - 1$ comparisons). [In fact, this is best possible.]
Αν το λύσετε με τη solution1 να εμφανίσετε τον πίνακα με τα N στοιχεία, min, max.
Αν το λύσετε με τη solution2 να εμφανίσετε τον πίνακα με τα N στοιχεία, τον A, B, min, max.
Παραδοτέο: p2e3.zip θα περιλαμβάνει όλα τα .java αρχεία & class αρχεία και δεδομένα I/O(*). Δοκιμάστε για περισσότερα από ένα σύνολα δεδομένων (εξετάστε όλες τις δυνατές περιπτώσεις)

(*) Για όλες τις παραπάνω ασκήσεις θα συμπεριλάβετε στο zip και τα αρχεία δεδομένων I/O. Ελέγξτε τον κώδικά σας με ανακατεύθυνση (redirecting from a file to standard input & redirecting from standard output to a file). Η ονομασία των αρχείων δεδομένων θα είναι in-p2ex.txt & out-p2ex.txt, όπου x: ο αύξων αριθμός της άσκησης. Ενδεχόμενα όπου χρειάζεται να δημιουργήσετε περισσότερα από ένα set δεδομένων εισόδου ή/και εξόδου.

Σε όλες τις κλάσεις που θα υλοποιήσετε θα πρέπει να συμπεριλάβετε στην αρχή της κλάσης αντίστοιχα και τα εξής:

```

/*****
 * Compilation: javac p2ex.java
 * Execution: java p2ex < in-p2ex.txt > out-p2ex.txt
 * Dependencies: StdIn.java StdOut.java Stack.java .....
 * Data files: in-p2ex.txt
 */
*****/

```