

## Άσκηση 2 (Προγραμματιστική)

Ημερομηνία Παράδοσης: 20 Δεκεμβρίου 2013

**Πρόβλημα 1 [100 μονάδες]** Σκοπός της άσκησης αυτής είναι να υλοποιήσετε δύο αλγορίθμους εύρεσης του διαμέσου στοιχείου ενός συνόλου θετικών ακεραίων αριθμών που σας δίνεται ως πίνακας. Θυμίζω ότι το διάμεσο στοιχείο ενός συνόλου  $n$  αριθμών είναι το στοιχείο τάξης  $\lceil \frac{n}{2} \rceil$ .

Οι δύο αλγόριθμοι που θα υλοποιήσετε περιγράφονται παρακάτω. Και στις δύο περιπτώσεις η είσοδος θεωρούμε ότι είναι ένας πίνακας  $A$  με  $n$  στοιχεία, με μη αρνητικές τιμές.

1. Τον πρώτο αλγόριθμο θα τον ονομάσουμε COUNTINGSELECT. Βασίζεται στον αλγόριθμο COUNTINGSORT και εκτελεί τα παρακάτω βήματα.
  - (α') Υπολογίζουμε το μέγιστο στοιχείο  $M$  του πίνακα  $A$ .
  - (β') Ταξινομούμε τα στοιχεία του  $A$  εκτελώντας τον αλγόριθμο COUNTINGSORT με είσοδο τον πίνακα  $A$  και μέγιστη τιμή για τα στοιχεία του  $A$  ίση με  $M$ .
  - (γ') Επιστρέφουμε το στοιχείο του (ταξινομημένου) πίνακα  $A$  τάξης  $\lceil \frac{n}{2} \rceil$ .
2. Ο δεύτερος αλγόριθμος που θα υλοποιήσετε είναι ο αλγόριθμος RANDOMIZEDSELECT, και θα πρέπει να εκτελεστεί στον πίνακα  $A$  ενώ το επιθυμητό στοιχείο είναι εκείνο τάξης  $\lceil \frac{n}{2} \rceil$ .

Στη συνέχεια για καθέναν από τους δύο αλγορίθμους θα πρέπει να γράψετε πρόγραμμα το οποίο θα διαβάζει τα στοιχεία του πίνακα  $A$  από αρχείο το οποίο θα δίνεται στο πρόγραμμα από το command line. Το πρώτο στοιχείο του αρχείου θα είναι πάντα το πλήθος  $n$  των στοιχείων του  $A$ , ενώ στη συνέχεια το αρχείο περιέχει τους  $n$  ακεραίους που απαρτίζουν τα στοιχεία του πίνακα.

Τις υλοποιήσεις θα τις χρησιμοποιήσετε στη συνέχεια ως εξής: σε URL που θα βρείτε στην ιστοσελίδα του μαθήματος θα βρείτε ένα .zip αρχείο με δεδομένα. Τα αρχεία αυτά ονομάζονται  $nX-Y.cin$ , όπου  $X$  και  $Y$  είναι ακέραιοι. Το  $X$  αναφέρεται στο πλήθος των στοιχείων του πίνακα  $A$  που περιέχονται στο αρχείο, ενώ το  $Y$  είναι κάποιος αύξοντας αριθμός του αρχείου. Έτσι το αρχείο  $n10000-2.cin$  περιέχει 10000 αριθμούς (μετά τον αριθμό 10000 που είναι ο πρώτος αριθμός στο αρχείο), και έχει αύξοντα αριθμό 2.

Χρησιμοποιώντας την εντολή `time` του Unix/Linux, μπορείτε να υπολογίσετε το χρόνο που χρειάζεται για να τρέξει το πρόγραμμά σας το κάθε αρχείο εισόδου δεδομένων. Από το αποτέλεσμα της εντολής αυτής θα πρέπει να κρατήσετε κάθε φορά το κομμάτι `user`.

Βάσει των προγραμμάτων σας και των αρχείων δεδομένων θα πρέπει να φτιάξετε γραφική παράσταση του χρονικού κόστους κάθε αλγορίθμου σας συναρτήσει του  $n$ . Για τον υπολογισμό του χρονικού κόστους θα πρέπει να τρέξετε το πρόγραμμά σας σε όλα τα αρχεία με το ίδιο  $X$  και να υπολογίσετε το μέσο όρο των επί μέρους χρόνων. Αυτή η διαδικασία θα πρέπει να γίνει για κάθε τιμή του  $X$  στα αρχεία που θα βρείτε.

Τί θα παραδώσετε: πέραν του κώδικα σας, θα πρέπει να γράψετε ένα μικρό κείμενο στον υπολογιστή που να περιγράφει σύντομα τον κώδικά σας και που θα περιέχει σε πίνακα του χρόνου κάθε αλγορίθμου σε κάθε αρχείο δεδομένων, αλλά και τις δύο γραφικές σας παραστάσεις. Τί παρατηρείτε;

Τέλος το πρόγραμμά σας θα πρέπει να γραφτεί σε C ή C++. Για τον υπολογισμό του χρονικού κόστους του αλγορίθμου σας χρησιμοποιήστε το compiler option `"-O3"`. Ο κώδικας σας θα πρέπει να μεταγλωττίζεται και να τρέχει στα μηχανήματα Unix/Linux των εργαστηρίων του τμήματος.

Σύνολο μονάδων: 100