ΑΣΚΗΣΗ 3

Η άσκηση 3 βαθμολογείται με 1 μονάδα. Η άσκηση είναι ατομική.

Οι φοιτητές των οποίων ο αριθμός μητρώου τελειώνει σε 0 ή 1 θα επιλύσουν το (a) ερώτημα, οι φοιτητές των οποίων ο αριθμός μητρώου τελειώνει σε 2 έως και 4 θα επιλύσουν το (b) ερώτημα, οι φοιτητές των οποίων ο αριθμός μητρώου τελειώνει σε 5 ή 6 θα επιλύσουν το (c) ερώτημα και οι φοιτητές των οποίων ο αριθμός μητρώου τελειώνει σε 7 έως και 9 θα επιλύσουν το (d) ερώτημα.

Ανεβάστε στο eclass ένα αρχείο zip της μορφής «ΟΝΟΜΑ_ΕΠΩΝΥΜΟ_ΑΜ.zip» (π.χ. ΝΙΚΟΛΑΟΣ_ΠΛΟΣΚΑΣ_000.zip) που θα περιέχει τα αρχεία με κατάληξη .c ή/και .h (ΚΑΙ ΜΟΝΟ ΑΥΤΑ ΤΑ ΑΡΧΕΙΑ) με το πρόγραμμα που θα γράψετε και ένα αρχείο με κατάληξη .pdf (το πολύ τρεις σελίδες) με την επεξήγηση της υλοποίησής σας.

a. Χρησιμοποιήστε τα αρχεία insertionsort.c, selectionsort.c και quicksort.c από τον φάκελο στο eclass «Κώδικες/Αλγόριθμοι ταξινόμησης» και δημιουργήστε ένα αρχείο main.c που θα περιέχει τους κώδικες αυτών των τριών αλγορίθμων ταξινόμησης. Στη συνέχεια γράψτε ένα πρόγραμμα που θα εκτελεί τους αλγόριθμους ταξινόμησης για πίνακες διάστασης από 10,000 έως 100,000 με βήμα 10,000 και για κάθε διάσταση να δημιουργείτε 10 διαφορετικά στιγμιότυπα πινάκων με τυχαίους αριθμούς (συνάρτηση rand() για παραγωγή τυχαίου ακεραίου και συνάρτηση srand(time(NULL)) για αρχικοποίηση γεννήτριας τυχαίων αριθμών). Για κάθε εκτέλεση του αλγορίθμου να χρονομετρήσετε τον χρόνο εκτέλεσής του και εισάγετε έναν πίνακα στην αναφορά σας (αρχείο pdf) όπως αυτός που φαίνεται στο τέλος αυτού του εγγράφου. Δείξτε τα αποτελέσματα σε ένα γράφημα. Σχολιάστε τα αποτελέσματα. Δείξτε επίσης ένα screenshot με την εκτέλεση του προγράμματος.
b. Χρησιμοποιήστε τα αρχεία insertionsort.c, selectionsort.c και mergesort.c από τον φάκελο στο eclass «Κώδικες/Αλγόριθμοι ταξινόμησης» και δημιουργήστε ένα αρχείο main.c που θα περιέχει τους κώδικες αυτών των τριών αλγορίθμων ταξινόμησης. Στη συνέχεια γράψτε ένα πρόγραμμα που θα εκτελεί τους αλγόριθμους ταξινόμησης για πίνακες διάστασης από 10,000 έως 100,000 με βήμα 10,000 και για κάθε διάσταση να δημιουργείτε 10 διαφορετικά στιγμιότυπα πινάκων με τυχαίους αριθμούς (συνάρτηση rand() για παραγωγή τυχαίου ακεραίου και συνάρτηση srand(time(NULL)) για αρχικοποίηση γεννήτριας τυχαίων αριθμών). Για κάθε εκτέλεση του αλγορίθμου να χρονομετρήσετε τον χρόνο εκτέλεσής του και εισάγετε έναν πίνακα στην αναφορά σας (αρχείο pdf) όπως αυτός που φαίνεται στο τέλος αυτού του εγγράφου. Δείξτε τα αποτελέσματα σε ένα γράφημα. Σχολιάστε τα αποτελέσματα. Δείξτε επίσης ένα screenshot με την εκτέλεση του προγράμματος.
c. Χρησιμοποιήστε τα αρχεία insertionsort.c, quicksort.c και mergesort.c από τον φάκελο στο eclass «Κώδικες/Αλγόριθμοι ταξινόμησης» και δημιουργήστε ένα αρχείο main.c που θα περιέχει τους κώδικες αυτών των τριών αλγορίθμων ταξινόμησης. Στη συνέχεια γράψτε ένα πρόγραμμα που θα εκτελεί τους αλγόριθμους ταξινόμησης για πίνακες διάστασης από 10,000 έως 100,000 με βήμα 10,000 και για κάθε διάσταση να δημιουργείτε 10 διαφορετικά στιγμιότυπα πινάκων με τυχαίους αριθμούς (συνάρτηση rand() για παραγωγή τυχαίου ακεραίου και συνάρτηση srand(time(NULL)) για αρχικοποίηση γεννήτριας τυχαίων αριθμών). Για κάθε εκτέλεση του αλγορίθμου να χρονομετρήσετε τον χρόνο εκτέλεσής του και εισάγετε έναν πίνακα στην αναφορά σας (αρχείο pdf) όπως αυτός που φαίνεται στο τέλος αυτού του εγγράφου. Δείξτε τα

αποτελέσματα σε ένα γράφημα. Σχολιάστε τα αποτελέσματα. Δείξτε επίσης ένα screenshot με την εκτέλεση του προγράμματος.

d. Χρησιμοποιήστε τα αρχεία insertionsort.c, improved_quicksort και nonrec_mergesort.c από τον φάκελο στο eclass «Κώδικες/Αλγόριθμοι ταξινόμησης» και δημιουργήστε ένα αρχείο main.c που θα περιέχει τους κώδικες αυτών των τριών αλγορίθμων ταξινόμησης. Στη συνέχεια γράψτε ένα πρόγραμμα που θα εκτελεί τους αλγόριθμους ταξινόμησης για πίνακες διάστασης από 10,000 έως 100,000 με βήμα 10,000 και για κάθε διάσταση να δημιουργείτε 10 διαφορετικά στιγμιότυπα πινάκων με τυχαίους αριθμούς (συνάρτηση rand() για παραγωγή τυχαίου ακεραίου και συνάρτηση srand(time(NULL)) για αρχικοποίηση γεννήτριας τυχαίων αριθμών). Για κάθε εκτέλεση του αλγορίθμου να χρονομετρήσετε τον χρόνο εκτέλεσής του και εισάγετε έναν πίνακα στην αναφορά σας (αρχείο pdf) όπως αυτός που φαίνεται στο τέλος αυτού του εγγράφου. Δείξτε τα αποτελέσματα σε ένα γράφημα. Σχολιάστε τα αποτελέσματα. Δείξτε επίσης ένα screenshot με την εκτέλεση του προγράμματος.

ΟΔΗΓΙΕΣ ΓΙΑ ΟΛΑ ΤΑ ΕΡΩΤΗΜΑΤΑ

Για να χρονομετρήσετε τον χρόνο εκτέλεσης μιας συνάρτησης μπορείτε να χρησιμοποιήσετε τον παρακάτω κώδικα:

```
#include <time.h>
.
.
.
clock_t start, end;
double cpu_time_used;
start = clock();
/* κλήση συνάρτησης */
end = clock();
cpu_time_used = ((double) (end - start)) / CLOCKS_PER_SEC;
```

Πίνακας 1: Πίνακας για να χρησιμοποιήσετε στην αναφορά σας με σκοπό να δείξετε τον χρόνο εκτέλεσης για κάθε αλγόριθμο. ΠΡΟΣΟΧΗ: Κάθε αλγόριθμος τρέχει 10 φορές για κάθε διάσταση και σε κάθε κελί βάζετε τον μέσο όρο αυτών των εκτελέσεων. Στην τελευταία γραμμή βάζετε τον μέσο όρο από όλες τις εκτελέσεις. Οι χρόνοι αυτοί πρέπει να υπολογίζονται μέσα στον κώδικά σας.

Διάσταση\ Αλγόριθμος	Αλγόριθμος Α	Αλγόριθμος Β	Αλγόριθμος Γ
10,000			
20,000			
30,000			
…			
100,000			
Μέσος όρος			

Ημερομηνία παράδοσης: 16 Ιανουαρίου 2023

Προσοχή:

1. Θα υπάρχει προφορική εξέταση όλων των ασκήσεων στο τέλος του εξαμήνου.
2. Υπάρχει ειδικό πρόγραμμα που αναγνωρίζει ομοιότητες στους κώδικες. Περιπτώσεις με ομοιότητες στον κώδικα (ή/και με ελλιπή επεξήγηση στο αρχείο .pdf) δε θα βαθμολογούνται.
3. Αν ένας φοιτητής καταθέσει άλλο ερώτημα από αυτό που πρέπει να κάνει (π.χ. φοιτητής του οποίου ο αριθμός μητρώου τελειώνει σε 8, καταθέσει το ερώτημα (a)), τότε δε θα βαθμολογείται.