

K08 Δομές Δεδομένων και Τεχνικές Προγραμματισμού

Διδάσκων: Μανόλης Κουμπάρκης

Εαρινό Εξάμηνο 2022-2023

Εργασία 2

Ανακοινώθηκε την 1η Απριλίου 2023

Προθεσμία: 30 Απριλίου 2023 στις 23:59

1.6 μονάδες στις 10 του βαθμού στο μάθημα (Άριστα=290 μονάδες)

Προσοχή: Πριν διαβάσετε παρακάτω, διαβάστε παρακαλώ προσεκτικά τις οδηγίες υποβολής των ασκήσεων που βρίσκονται στην ιστοσελίδα <http://cgi.di.uoa.gr/~k08/homework.html>, ειδικά ότι αναφέρεται στο github και τα σχετικά αρχεία.

Απορίες: Αν έχετε απορίες σχετικά με την εργασία, ρωτήστε στο piazza και όχι στέλνοντας e-mail στον διδάσκοντα. Τέτοια e-mails ΔΕΝ θα λαμβάνουν απάντηση.

Κώδικας: Ο κώδικας που παρουσιάσαμε στις διαλέξεις του μαθήματος (Ενότητες 6-12) και θα χρειαστείτε για την εργασία αυτή βρίσκεται στο παρακάτω private repository του classroom του μαθήματος: <https://github.com/artioi-k08/2023-ergasia2>. Για να συνδεθείτε στο repository αυτό, θα πρέπει να χρησιμοποιήσετε το λινκ <https://classroom.github.com/a/u30u3ezR>.

Όταν συνδεθείτε θα μπορείτε να δείτε το προσωπικό σας repository <https://github.com/artioi-k08/2023-ergasia2-<github-your-username>> στο οποίο θα δουλέψετε για την Εργασία 2.

Μετά τη σύνδεση σας, στο προσωπικό σας repository, θα βρείτε τον κώδικα που καλύπτει τις Ενότητες 6-12 του μαθήματος οργανωμένο σε κατάλληλους φακέλους. Θα βρείτε επίσης και ένα φάκελο **solutions-ergasia2** με υπο-φακέλους **question1, question2, ..., question6** στους οποίους θα πρέπει να γράψετε τον κώδικα ή τις απαντήσεις σας για τις 6 ασκήσεις αυτής της εργασίας που θα

βρείτε παρακάτω. Παρακαλώ τηρείστε ευλαβικά αυτήν την οργάνωση αλλιώς θα χάσετε 20% του συνολικού βαθμού κατά την βαθμολόγηση. Για τα θεωρητικά ερωτήματα, οι απαντήσεις πρέπει να είναι σε ένα αρχείο τύπου pdf.

Κύριο πρόγραμμα: Σε όλες τις παρακάτω προγραμματιστικές ασκήσεις θα πρέπει να υλοποιήσετε και ένα κύριο πρόγραμμα (συνάρτηση `main`) το οποίο θα διαβάζει τα δεδομένα εισόδου, θα επιδεικνύει τη λειτουργικότητα της συνάρτησης σας για κατάλληλα επιλεγμένες εισόδους, και θα πείθει τον βαθμολογητή ώστε να σας βαθμολογήσει με τον υψηλότερο δυνατό βαθμό. Επίσης, τα προγράμματα σας δεν θα πρέπει να έχουν `memory errors` ή `leaks` (θα ελεγχθούν με τη χρήση `valgrind`).

1. Θεωρήστε το πρόγραμμα που βρίσκεται στο αρχείο `insertion-sort.c` που βρίσκεται στο repository σας (φάκελος `insertion-sort-code`). Το πρόγραμμα αυτό ταξινομεί ένα πίνακα ακεραίων με τη μέθοδο της ταξινόμησης με εισαγωγή. Ποια είναι η χρονική υπολογιστική πολυπλοκότητα χειρότερης περίπτωσης της συνάρτησης `sort`; Ποια είναι η αντίστοιχη πολυπλοκότητα της `main`; Υποθέστε ότι η κλήση `rand()` εκτελείται σε σταθερό χρόνο. Να εξηγήσετε με λεπτομέρεια τις απαντήσεις σας.

(10 μονάδες)

2. Υποθέστε ότι έχουμε 10 αλγόριθμους A, B, Γ, Δ, E, Z, H, Θ, I και K με τις παρακάτω υπολογιστικές πολυπλοκότητες χρόνου (παραλείπουμε το $O()$).

A. $2000n$ B. $50n + \log n$ Γ. $2^{3000n \log n}$ Δ. $2^{300 \log n}$ E. n^4
Z. $6n^4 + n$ H. $n \log n 2^{n+5}$ Θ. 2^{2^n} I. $2^{2^{\log n}}$ K. $2^{2^{n+\log n}}$

Να ταξινομήσετε τους αλγόριθμους από τον καλύτερο στον χειρότερο με βάση την υπολογιστική πολυπλοκότητα τους. Να δώσετε λεπτομερώς όσους μαθηματικούς υπολογισμούς χρειάζονται για να τεκμηριώσετε την απάντησή σας.

(10 μονάδες)

3. Δώστε μια ακολουθία 10 κλειδιών (χρησιμοποιήστε τα γράμματα A έως K) η οποία, όταν τα κλειδιά εισάγονται με τη σειρά αυτή σε ένα αρχικά άδαιο δυαδικό δένδρο αναζήτησης με τη μέθοδο της εισαγωγής στη ρίζα (σελίδες 81-90 των διαφανειών της Ενότητας 9), απαιτείται μέγιστος αριθμός

συγκρίσεων για να χτιστεί το δένδρο. Να δώσετε αυτό τον αριθμό συγκρίσεων.

(10 μονάδες)

4. Στην άσκηση αυτή θα επεκτείνετε την υλοποίηση του αφαιρετικού τύπου δεδομένων Πίνακας Συμβόλων (Symbol Table) με δένδρα δυαδικής αναζήτησης η οποία παρουσιάζεται στις διαφάνειες της Ενότητας 9.
 - Να οργανώσετε τον κώδικα του αρχείου `symbol-table.c` που βρίσκεται στο repository σας (φάκελος `symbol-table-code`) σε ένα module της C το οποίο υλοποιεί τον αφαιρετικό τύπο δεδομένων Πίνακας Συμβόλων με το interface που δίνεται στη σελίδα 5 των σχετικών διαφανειών. Μπορείτε να χρησιμοποιήσετε και να επεκτείνετε τη συνάρτηση `main` της σελίδας 65 των διαφανειών για να επιδείξετε τη λειτουργία του πίνακα συμβόλων με κατάλληλες κλήσεις στις συναρτήσεις που προσφέρονται από το interface. Είναι δική σας ευθύνη να επεκτείνετε τον κώδικα που σας δίνεται, ώστε να δουλεύουν πλήρως οι διάφορες συναρτήσεις. Για παράδειγμα, θα χρειαστεί να επεκτείνετε κάποιες συναρτήσεις που δεν ενημερώνουν το πεδίο `N` του struct `STnode` ώστε να το κάνουν.
 - Να προσθέσετε στο interface και να υλοποιήστε τη συνάρτηση `void STdeleteAll(Item)` που θα είναι μια νέα έκδοση της συνάρτησης `STdelete` (σελίδα 108 των διαφανειών της Ενότητας 9) η οποία διαγράφει όλα τα στοιχεία του πίνακα συμβόλων (δηλαδή, όλους τους κόμβους του δένδρου) που έχουν κλειδιά ίσα με το δοσμένο κλειδί. Να προσθέσετε κώδικα στη `main` η οποία να δείχνει τη λειτουργικότητα της συνάρτησης σας.
 - Να επεκτείνετε το λογισμικό που αναπτύξατε στα παραπάνω δύο ερωτήματα ώστε να μπορεί να χρησιμοποιηθεί στην περίπτωση που θέλουμε ο πίνακας συμβόλων να λειτουργεί σαν μητρώο των φοιτητών του τμήματος μας. Υποθέστε ότι σε κάθε `item` τώρα δεν αποθηκεύουμε ένα ακέραιο αλλά τις εξής πληροφορίες: το κλειδί που είναι ο αριθμός μητρώου του φοιτητή, το όνομα του και το επώνυμο του.

- Παρατηρήστε ότι ο κώδικας που δώσαμε στο μάθημα, και τον οποίο χρησιμοποιήσατε και επεκτείνετε στα παραπάνω ερωτήματα, μας επιτρέπει να έχουμε ένα μόνο πίνακα συμβόλων. Να υλοποιήσετε μια νέα έκδοση του λογισμικού των παραπάνω ερωτημάτων που να επιτρέπει τη δημιουργία πολλών πινάκων συμβόλων και την χρήση τους από ένα κύριο πρόγραμμα που θα επιδεικνύει τη λειτουργικότητα του λογισμικού σας. Σε αυτό το ερώτημα καλείστε να υλοποιήσετε unit tests για την νέα έκδοση του πίνακα συμβόλων.

Σε όλα τα παραπάνω ερωτήματα, θα πρέπει ο κώδικας των modules που θα αναπτύξετε να κάνει την καλύτερη δυνατή απόκρυψη πληροφορίας.

(20+10+10+50=90 μονάδες)

5. Να υλοποιήσετε τον αφαιρετικό τύπο δεδομένων Ουρά Προτεραιότητας χρησιμοποιώντας σωρούς μεγίστων όπως δείξαμε στην Ενότητα 8 των διαφανειών. Με βάση τον κώδικα των διαλέξεων, να υλοποιήσετε την παρακάτω διεπαφή που σας επιτρέπει να έχετε παραπάνω από μια ουρές προτεραιότητας:

```
typedef struct priority_queue{
    int Count;
    Item *ItemArray;
}PriorityQueue;
typedef PriorityQueue *PQPointer;
PQPointer QUEUEInit(int maxN);
int QUEUEempty(PQPointer);
void QUEUEput(PQPointer, Item);
Item QUEUEget(PQPointer);
```

Ο τύπος Item μπορεί να είναι int. Να γράψετε και ένα κύριο πρόγραμμα το οποίο επιδεικνύει τη συμπεριφορά της ουράς προτεραιότητας. Μαζί με το κύριο πρόγραμμα, καλείστε να υλοποιήσετε unit tests για τις συναρτήσεις της

διεπαφής. Μπορείτε να χρησιμοποιήσετε κώδικα από την Ενότητα 8 των διαφανειών.

(50 μονάδες)

6. Να υλοποιήσετε τον αφαιρετικό τύπο δεδομένων Δένδρο (2,4) που παρουσιάσαμε στην Ενότητα 11 των διαφανειών. Να χρησιμοποιήσετε την παρακάτω διεπαφή:

```
typedef struct t24 Tree24;
struct t24{
    int Count;
    Tree24 *parent;
    Item items[3];
    Tree24 *children[4];
    int N[4];
};
void init();
int count();
void insert(Item);
Item search(Key);
void delete(Item);
Item select(int);
void sort(void (*visit)(Item));
```

Όπως μπορείτε να δείτε η διεπαφή αυτή είναι η ίδια με την διεπαφή για τον πίνακα συμβόλων της άσκησης 4 παραπάνω. Να γράψετε και ένα κύριο πρόγραμμα το οποίο επιδεικνύει τη συμπεριφορά των παραπάνω συναρτήσεων. **ΔΕΝ** απαιτείται η υλοποίηση unit tests σε αυτή την άσκηση.

(120 μονάδες)

Καλή Επιτυχία!