

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Εργασία 2023-2024

1. Γενικά

Στόχος της εργασίας αυτής είναι η δημιουργία μίας βιβλιοθήκης η οποία θα περιλαμβάνει μερικές χρήσιμες δομές δεδομένων υλοποιημένες σε C++. Η ενασχόληση με την εργασία θα σας προσφέρει χρήσιμες γνώσεις γύρω από το θέμα ενώ παράλληλα θα σας δώσει την ευκαιρία να ενισχύσετε τις ικανότητές σας στον προγραμματισμό. Η εργασία θα εκπονηθεί σε ομάδες των δύο ατόμων. Η υποβολή της εργασίας θα γίνει στο elearning και η προθεσμία υποβολής είναι η **Κυριακή 23 Ιουνίου 2024**. Η εργασία λαμβάνει το **50% της βαθμολογίας** στο θεωρητικό κομμάτι των Δομών Δεδομένων. Το υπόλοιπο 50% αφορά στις τελικές εξετάσεις. Η εργασία θα υλοποιηθεί σε ομάδες δύο ατόμων.

2. Περιγραφή

Στη βιβλιοθήκη που θα υλοποιήσετε πρέπει να υποστηρίζονται οι ακόλουθες δομές δεδομένων: **σωρός ελαχίστων, σωρός μεγίστων, δυαδικό δένδρο αναζήτησης AVL, γράφημα χωρίς κατευθύνσεις αλλά με βάρη στις ακμές, πίνακας κατακερματισμού**. Για κάθε δομή, θα πρέπει να υπάρχει ένα αρχείο .h το οποίο θα πρέπει να γίνεται #include από το πρόγραμμα που θέλει να χρησιμοποιήσει την αντίστοιχη δομή. Επίσης, για κάθε δομή θα πρέπει να υπάρχει τουλάχιστον ένα .cpp αρχείο το οποίο υλοποιεί την αντίστοιχη δομή.

Το πρόγραμμά σας δε θα διαβάζει τίποτε από το πληκτρολόγιο και επίσης δε θα τυπώνει τίποτε στην οθόνη. Όταν ξεκινά το πρόγραμμα, θα πρέπει να διαβάζει το αρχείο **commands.txt** το οποίο περιέχει μία εντολή σε κάθε γραμμή. Το πρόγραμμά σας θα πρέπει να εκτελεί την κάθε εντολή ανάλογα με το περιεχόμενο και να δίνει την έξοδο στο αρχείο **output.txt**. Οι εντολές που μπορεί να περιέχει είναι οι εξής:

BUILD MINHEAP filename	Κατασκευή σωρού ελαχίστων διαβάζοντας τα στοιχεία από το αρχείο filename.
BUILD MAXHEAP filename	Κατασκευή σωρού μεγίστων διαβάζοντας τα στοιχεία από το αρχείο filename.
BUILD AVL TREE filename	Κατασκευή δυαδικού δένδρου αναζήτησης AVL διαβάζοντας τα στοιχεία από το αρχείο filename
BUILD GRAPH filename	Κατασκευή γραφήματος διαβάζοντας τα στοιχεία από το αρχείο filename.
BUILD HASHTABLE filename	Κατασκευή πίνακα κατακερματισμού διαβάζοντας τα στοιχεία από το αρχείο filename
GETSIZE MINHEAP (ομοίως για MAXHEAP, AVL TREE, GRAPH, HASHTABLE)	Επιστρέφει το πλήθος των στοιχείων του σωρού ελαχίστων. Για την περίπτωση του γράφου επιστρέφονται δύο τιμές, το πλήθος των κορυφών και το πλήθος των ακμών.
FINDMIN MINHEAP	Εύρεση του ελάχιστου στοιχείου από το σωρό ελαχίστων.
FINDMAX MAXHEAP	Εύρεση του μέγιστου στοιχείου από το σωρό μεγίστων.
FINDMIN AVL TREE	Εύρεση του ελάχιστου στοιχείου από το δένδρο AVL.
SEARCH AVL TREE number	Αναζήτηση του στοιχείου number στο δένδρο AVL. Αν το στοιχείο υπάρχει, επιστρέφεται η συμβολοσειρά SUCCESS αλλιώς FAILURE.
SEARCH HASHTABLE number	Αναζήτηση του στοιχείου number στον πίνακα

	κατακερματισμού. Αν το στοιχείο υπάρχει, επιστρέφεται η συμβολοσειρά SUCCESS αλλιώς FAILURE.
COMPUTESHORTESTPATH GRAPH number1 number2	Υπολογίζει το κόστος της ελάχιστης διαδρομής από τον κόμβο number1 στον κόμβο number2.
COMPUTESPANNINGTREE GRAPH	Υπολογίζει το κόστος των ελάχιστου εκτεινόμενου δένδρου του γραφήματος και το επιστρέφει.
FINDCONNECTEDCOMPONENTS GRAPH	Υπολογίζει το πλήθος των συνεκτικών συνιστωσών του γραφήματος και επιστρέφει τον αριθμό αυτό.
INSERT MAXHEAP number (ομοίως για MAXHEAP, AVL TREE, HASHTABLE)	Εισάγει τον αριθμό number στο σωρό ελαχίστων.
INSERT GRAPH number1 number 2	Εισάγει την ακμή <node1, node2, w> στο γράφημα αν η ακμή δεν υπάρχει.
DELETEMIN MINHEAP	Διαγραφή ελαχίστου από MINHEAP.
DELETEMAX MAXHEAP	Διαγραφή μεγίστου από MAXHEAP.
DELETE AVL TREE number	Διαγραφή στοιχείου από AVL.
DELETE GRAPH number1 number2	Διαγραφή της ακμής <number1, number2> από το γράφημα εάν υπάρχει.

Σημείωση: να μην υλοποιηθεί η διαγραφή στο HASHTABLE.

Η κάθε εντολή του αρχείου `commands.txt` θα πρέπει να εκτελεστεί χρησιμοποιώντας εντίστοιχες μεθόδους από τις δομές δεδομένων που θα υλοποιήσετε. **Η ανάγνωση του αρχείου θα πρέπει να γίνεται μέσα στη `main()` ή όπου αλλού είναι βολικό.**

Το αποτέλεσμα της κάθε εντολής θα πρέπει να εγγράφεται στο μοναδικό αρχείο `output.txt` μαζί με μαζί με το χρόνο εκτέλεσης. Σημειώνεται ότι η σειρά των εντολών στο αρχείο `commands.txt` μπορεί να είναι τυχαία. Ωστόσο θα είναι πάντα σωστό το αρχείο, με την έννοια ότι δε θα περιέχει γραμματικά λάθη και επίσης για να υπάρχει εντολή χρήσης μίας δομής πρέπει πρώτα η δομή να χτιστεί με την εντολή `BUILD`.

Το `input.txt` θα περιέχει σε κάθε γραμμή έναν αριθμό ≥ 0 , ενώ για την περίπτωση του γραφήματος, σε κάθε γραμμή θα υπάρχουν τρεις αριθμοί `node1 node2 weight` όπου `node1` και `node2` είναι οι κωδικοί των κορυφών και `weight` το βάρος της ακμής που εισάγεται.

3. Απαιτήσεις

Ο κώδικας θα πρέπει να είναι δικό σας και είναι επαρκώς σχολιασμένος (τα σχόλια είτε στην ελληνική γλώσσα είτε στην αγγλική αλλά όχι `greeklish`). Για να βαθμολογηθεί η εργασία θα πρέπει ο κώδικας να περνάει τουλάχιστον τη φάση της μεταγλώττισης χωρίς σφάλματα. Διαφορετικά δε θα βαθμολογείται καθόλου. Τα τελικά παραδοτέα σας είναι:

- Πηγαίος κώδικας με όλα τα `.cpp` και όλα τα `.h` αρχεία συμπεριλαμβανομένης και της `main()`.
- Τεχνική έκθεση όπου θα αναφέρονται λεπτομέρειες για τις υλοποιήσεις σας. Η έκθεση θα πρέπει να περιέχει οπωσδήποτε και το `pdf` αρχείο.

Εάν κάποιο από τα παραδοτέα απουσιάζει, η εργασία δε θα βαθμολογείται. Όλα τα απαραίτητα αρχεία θα πρέπει να συγκεντρωθούν σε ένα αρχείο με όνομα `AEM1-AEM2.zip` (ή `rar`, ή ότι άλλο θέλετε) όπου `AEM1` και `AEM2` είναι οι αριθμοί μητρώου των μελών της ομάδας.

Καλή επιτυχία