

# ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

## Εργασία 2019-2020

### 1. Γενικά

Στόχος της εργασίας αυτής είναι η υλοποίηση σε γλώσσα C++ μερικών βασικών δομών δεδομένων, με σκοπό την αποθήκευση πληροφορίας κειμένου. 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 7 Ιουνίου 2020** (δε θα υπάρξει καμία παράταση). 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 των δύο ατόμων. Θα γίνουν δεκτές και ατομικές εργασίες. Άτομα που είχαν παραδώσει εργασία τις προηγούμενες χρονιές μπορούν να κρατήσουν το βαθμό τους στην εργασία. Επίσης πρέπει να γνωρίζετε ότι για περάσετε το μάθημα θα πρέπει να έχετε τη βάση και στην εργασία αλλά και στην εξέταση.

### 2. Περιγραφή

Οι δομές που πρέπει να υλοποιηθούν είναι: 1) απλό δυαδικό δένδρο αναζήτησης, 2) δυαδικό δένδρο αναζήτησης τύπου AVL και 3) πίνακας κατακερματισμού με ανοικτή διεύθυνση. Οι δομές αυτές θα πρέπει να αποθηκεύουν τις διαφορετικές λέξεις του κειμένου και το πλήθος εμφανίσεων της κάθε λέξης μέσα στο κείμενο. Υποθέτουμε ότι μία λέξη λ1 είναι μικρότερη από μία λέξη λ2, όταν η λ1 λεξικογραφικά βρίσκεται πριν από τη λέξη λ2. Για παράδειγμα “that” < “this”, “apple” < “cat”, “over” < “under” κλπ. Το κείμενο πάνω στο οποίο θα δουλέψετε είναι από το Gutenberg Project (<https://www.gutenberg.org/>) και περιέχει γνωστά έργα παγκόσμιας λογοτεχνίας. Το αρχείο με το οποίο θα εργαστείτε θα αναρτηθεί στο elearning μαζί με την εκφώνηση της εργασίας. Ωστόσο, μπορείτε να χρησιμοποιήσετε στις δοκιμές σας και μικρότερα αρχεία ώστε να μπορείτε να ελέγχετε την ορθότητα της υλοποίησής σας.

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. Όταν το πρόγραμμα εκτελεστεί θα πρέπει αρχικά να κατασκευάσει τις τρεις δομές δεδομένων εισάγοντας τις λέξεις μία προς μία καθώς διαβάζουμε τις γραμμές του αρχείου. Στη συνέχεια, θα πρέπει από το αρχείο κειμένου να επιλεγεί με τυχαίο τρόπο ένα σύνολο Q από π.χ. 1000 λέξεις (όχι κατ’ ανάγκη διαφορετικές) οι οποίες θα χρησιμοποιηθούν για να εκτελέσουμε αναζητήσεις στις δομές που έχουμε υλοποιήσει. Στη συνέχεια αναζητούμε όλες τις λέξεις του συνόλου Q σε κάθε δομή και επιστρέφουμε τον αντίστοιχο συνολικό χρόνο εκτέλεσης (πόσο χρόνο χρειάστηκε η δομή να απαντήσει στα ερωτήματα) καθώς και πόσες φορές εμφανίζεται η κάθε λέξη. Είναι προφανές ότι για κάθε λέξη που αναζητούμε, οι τρεις δομές θα πρέπει να δώσουν ίδιο αποτέλεσμα ως προς το πλήθος των εμφανίσεων αλλά διαφορετικό χρόνο εκτέλεσης. Η κατασκευή των δομών και η αναζήτηση θα γίνεται από τη `main()` η οποία πρέπει να καλέσει τις κατάλληλες μεθόδους από τις κλάσεις που υλοποιούν τις δομές.

Επίσης, το πρόγραμμά σας θα πρέπει να υποστηρίζει για τις δομές των δένδρων: εισαγωγή, διαγραφή, αναζήτηση, `inorder`, `preorder`, `postorder`. Για τον πίνακα κατακερματισμού απαιτείται μόνο εισαγωγή και αναζήτηση (**να μην υλοποιηθεί η διαγραφή**).

Στην αναφορά που θα παραδώσετε θα πρέπει να εξηγήσετε τον τρόπο σχεδιασμού και υλοποίησης της εφαρμογής σας και γενικά να δώσετε τις πληροφορίες που θεωρείτε απαραίτητες. Επίσης, ο κώδικάς σας θα πρέπει να περιέχει επαρκή σχόλια (είτε στα ελληνικά είτε στα αγγλικά αλλά όχι σε greeklish!).

### 3. Απαιτήσεις - Παραδοτέα

Ο κώδικας θα πρέπει να είναι δικός σας και να είναι επαρκώς σχολιασμένος. **Δεν επιτρέπεται η χρήση της βιβλιοθήκης STL όπως και άλλων βιβλιοθηκών που έχουν υλοποιημένες διάφορες δομές δεδομένων.** 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. Φροντίστε ώστε η έκθεση να υπάρχει και σε μορφή pdf εκτός από doc ή odt.

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**