ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ
Εργασία 2020-2021

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 20 Ιουνίου 2021**. Η εργασία λαμβάνει το **50% της βαθμολογίας** στο θεωρητικό κομμάτι των Δομών Δεδομένων (το οποίο λαμβάνει 60% της συνολικής βαθμολογίας του μαθήματος, ενώ το εργαστηριακό μέρος λαμβάνει 40%). Το **υπόλοιπο 50%** αφορά στις τελικές εξετάσεις. Η εργασία θα υλοποιηθεί σε ομάδες των δύο ατόμων. Επίσης πρέπει να γνωρίζετε ότι για περάσετε το μάθημα θα πρέπει να έχετε **τη βάση στην εξέταση**. Εάν δεν εκπονήσετε την εργασία, χάνετε το 30% του τελικού βαθμού.

2. Περιγραφή

Οι δομές που πρέπει να υλοποιηθούν στα πλαίσια της εργασίας είναι:

1) *αταξινόμητος πίνακας*,
2) *ταξινομημένος πίνακας*,
3) *απλό δυαδικό δένδρο αναζήτησης*,
4) *δυαδικό δένδρο αναζήτησης τύπου AVL* και
5) *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. Στο elearning υπάρχει διαθέσιμο ένα **μικρό αρχείο για να δουλέψετε**. Το αρχείο με το οποίο θα πρέπει να κάνετε τις τελικές δοκιμές σας είναι αρκετά μεγαλύτερο και το μέγεθός του είναι περίπου 2.2GBytes (συμπιεσμένο περίπου 850MBytes). Το αρχείο είναι διαθέσιμο στον ακόλουθο σύνδεσμο:

https://aristotleuniversity-my.sharepoint.com/:u:/g/personal/papadopo_office365_auth_gr/EZvtISJTl75LiUguL5VZiuUBtWvb1JJxotS92i4_3Uevlw?e=uv2KBj

Η εργασία σας θα δοκιμαστεί με αρχεία μεγέθους αντίστοιχου με το προηγούμενο. Άρα, θα πρέπει να κάνετε τις απαραίτητες δοκιμές ώστε ο κώδικάς σας να λειτουργεί σωστά.

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 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 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: εισαγωγή, διαγραφή, αναζήτηση, για τις δενδρικές δομές επιπλέον inorder, preorder, postorder. Για τον πίνακα κατακερματισμού απαιτείται μόνο εισαγωγή και αναζήτηση (**να μην υλοποιηθεί η διαγραφή**).

Σημειώνεται επίσης ότι όταν δουλεύουμε με κείμενο, πολλές φορές εκτελούμε κάποιου είδους προεπεξεργασία. Για παράδειγμα, συχνά μετατρέπουμε κεφαλαία σε πεζά γράμματα (γιατί π.χ. η λέξη “This” είναι ίδια με τη λέξη “this”) και αφαιρούμε σημεία στίξης. Αυτές οι δύο λειτουργίες θα πρέπει να εκτελούνται από τον κώδικά σας κατά την ανάγνωση του αρχείου εισόδου.

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 δομέων δεδομένων από τη βιβλιοθήκη STL (vector, unordered_map, κλπ) όπως και η χρήση 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 ΑΕΜ1 και ΑΕΜ2 είναι οι αριθμοί μητρώου των μελών της ομάδας. Στην αναφορά να εμφανίζονται τα ονόματα και τα ΑΕΜ των μελών της ομάδας.

Καλή επιτυχία