

Τεχνική Έκθεση Εργασίας στις Δομές Δεδομένων

Η εργασία αποτελείται από τα αρχεία .cpp και .h των κλάσεων NonSortedArray, SortedArray, BinarySearchTree, Node, AVLTree, NodeAVL και Hashtable που υλοποιούν τις αντίστοιχες δομές.

- **Main().** Στη main γίνεται η επεξεργασία και η τροποποίηση του αρχείου κειμένου που δέχονται ως είσοδο οι δομές, επιλέγονται τυχαία οι λέξεις που θα αναζητηθούν και ξεκινάει η διαδικασία του χτισίματος στις δομές δυαδικό δένδρο, δένδρο AVL και hashtable. Τέλος στην main δηλώνονται δύο μεταβλητές filename και search_size για την ευκολότερη αλλαγή του αρχείου εισόδου αλλά και του αριθμού των ζευγών λέξεων που θα αναζητηθούν.
- **NonSortedArray.** Η NonSortedArray υλοποιεί έναν μη ταξινομημένο πίνακα. Στον constructor καλείται η συνάρτηση BuildArray() η οποία εισάγει γραμμικά τα ζεύγη λέξεων που διαβάζει από το αρχείο. Όποτε τελειώνει ο χώρος που δεσμεύτηκε για τον πίνακα καλείται η συνάρτηση addMemory η οποία δημιουργεί έναν καινούργιο πίνακα με διπλάσιο χώρο και αντιγράφει τα δεδομένα του προηγούμενου πίνακα σε αυτόν. Τέλος διαγράφεται ο παλιός πίνακας και πλέον η μεταβλητή string *array παίρνει την τιμή της διεύθυνσης του καινούργιου πίνακα. Η αναζήτηση στον πίνακα γίνεται γραμμικά μέσω της συνάρτησης searchArray που καλείται από την main και δέχεται ως ορίσματα έναν πίνακα με τα ζεύγη λέξεων που πρέπει να βρεθούν και το μέγεθος του. Τέλος ο destructor διαγράφει τον δείκτη του array.
- **SortedArray.** Υλοποιεί την δομή του ταξινομημένου πίνακα. Κληρονομεί την κλάση NonSortedArray και καλεί την συνάρτηση BuildArray για την αρχική εισαγωγή των δεδομένων. Στην συνέχεια γίνεται η ταξινόμηση με τον αλγόριθμο quicksort μέσω των συναρτήσεων quicksort() και partition() όπου η quicksort λειτουργεί αναδρομικά με βάση τα χωρίσματα που επιστρέφει κάθε φορά η partition(). Για την αναζήτηση υλοποιήθηκε ο αλγόριθμος δυαδικής αναζήτησης ο οποίος καλείται από την main μέσω της συνάρτησης binarySearch(). Τέλος ο destructor διαγράφει τον δείκτη του array.
- **BinarySearchTree και Node.** Υλοποιούν ένα δυαδικό δένδρο αναζήτησης. Η Node αναπαριστά τον κάθε κόμβο του δένδρου και έχει ως μεταβλητές δύο δείκτες Node* για το δεξί και το αριστερό παιδί του

κόμβου ,που αρχικοποιούνται στην τιμή NULL μέχρι να εισαχθεί ένας κόμβος σε αυτή τη θέση, την συμβολοσειρά string words όπου αποθηκεύεται το ζεύγος λέξεων του κόμβου και την μεταβλητή count που μετράει τον αριθμό εμφανίσεων του ζεύγους λέξεων. Η BinarySearchTree εισάγει τους κόμβους στα κατάλληλα σημεία σύμφωνα με τους κανόνες του δυαδικού δένδρου(αριστερά κάθε κόμβου οι μικρότερες τιμές και δεξιά οι μεγαλύτερες) και αποθηκεύει την ρίζα του δένδρου. Η συνάρτηση που υλοποιεί αυτή την δυνατότητα είναι η InsertNode η οποία δέχεται από την main τον καινούργιο κόμβο προς εισαγωγή και τον τοποθετεί στο σημείο που πρέπει. Η αναζήτηση στον κόμβο γίνεται με την συνάρτηση findData() που καλεί η main και δέχεται ως ορίσματα έναν πίνακα με τα ζεύγη λέξεων που πρέπει να βρεθούν και το μέγεθος του. Τέλος ο destructor της BinarySearchTree διαγράφει τον δείκτη του root και ο destructor του Node τους δείκτες left και right.

- Hash table . Λόγω λάθους και βιασύνης υλοποιήθηκε κατακερματισμός με αλυσίδες. Παρ 'όλα αυτά για να μην μείνει κενή η υλοποίηση συμπεριλήφθηκε και αυτή η δομή. Η αρχή του χτισίματος ξεκινάει από την main με την συνάρτηση createTable .Η κλάση List υλοποιεί τις αλυσίδες που βρίσκονται σε κάθε θέση του πίνακα. Αποτελείται από έναν δείκτη στην διεύθυνση του επόμενου αντικειμένου λίστας το ζεύγος λέξεων που αποθηκεύεται και τον αριθμό εμφανίσεων του. Το μέγεθος του πίνακα καθορίζεται ως ένας πρώτος αριθμός για την καλύτερη λειτουργία του αλγόριθμου κατακερματισμού. Η αναζήτηση γίνεται από την συνάρτηση searchHashtable η οποία δέχεται έναν πίνακα συμβολοσειρών και το μέγεθος του. Για κάθε συμβολοσειρά υπολογίζεται η θέση που μπορεί να βρεθεί στον πίνακα με βάση τον αλγόριθμο κατακερματισμού που χρησιμοποιήθηκε στην κατασκευή του πίνακα και στη συνέχεια γίνεται αναζήτηση στην λίστα.

Ο χρόνος κατασκευής της κάθε δομής ,αναζήτησης σε αυτή και οι λέξεις που αναζητήθηκαν αποθηκεύονται στο αρχείο results.txt

Βιβλιογραφία:

- <https://www.geeksforgeeks.org/binary-search/>
- <https://www.geeksforgeeks.org/quick-sort/>