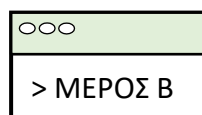


3η Εργασία Δομές Δεδομένων - Report

Μιλτιάδης Τσόλκας : 3200213

Χαράλαμπος Καρακώστας : 3200065



- **void insert(String w):** Η μέθοδος αυτή εισάγει κόμβο στο δέντρο αν δεν υπάρχει ήδη αυτός ο κόμβος στο δέντρο, αλλιώς ενημερώνει το πεδίο που αφορά τις φορές εμφάνισης μιας λέξης. Η λειτουργία της είναι αναδρομική και με κάθε προσθήκη κόμβου ενημερώνεται ανάλογα το πεδίο subTreeSize.
- **WordFreq search(String w):** Η μέθοδος αυτή αναζητά λέξη στο δέντρο και αν υπάρχει επιστρέφει το αντίστοιχο στοιχείο WordFreq, αλλιώς NULL. Επιπλέον, αν η συχνότητα της λέξης που αναζητάμε είναι μεγαλύτερη της μέσης συχνότητας (από την getMeanFrequency()), τότε με χρήση περιστροφών (μέθοδοι rotateRight και rotateLeft) θα φέρνει τη λέξη αυτή στη ρίζα του δέντρου, μέσω της μεθόδου insertHead. Μετά τις αλλαγές στην διάταξη του δέντρου, ενημερώνονται ανάλογα τα πεδία subTreeSize.
- **void remove(String w):** Η μέθοδος αυτή αφαιρεί κόμβο από το δέντρο, αν δεν είναι άδαιο, αφού πρώτα τον βρει με αναδρομική αναζήτηση. Μετά την αφαίρεση του κόμβου αντικαθίσταται ως «παιδί» αυτού του «γονιού» ο αμέσως επόμενος κόμβος αν υπάρχει. Αφού αφαιρεθεί ο κόμβος ενημερώνονται ανάλογα τα πεδία subTreeSize.
- **void load(String filename):** Η μέθοδος αυτή κάνει load το αρχείο txt στο ΔΔΑ. Οι λέξεις του txt ελέγχονται με συνθήκες σύμφωνα με τις απαιτήσεις της εκφώνησης (σχετικά με το τι πρέπει να γίνεται με τα σημεία στίξης κλπ.) και αν η λέξη δεν περιλαμβάνεται στην λίστα των stopWords πραγματοποιούμε insert στο δέντρο. Ελέγχουμε πριν από κάθε load αν έχει προστεθεί κάποια λέξη στις stopWords η οποία είχε προηγουμένως φορτωθεί στο δέντρο και αν ναι την αφαιρούμε. Θεωρήσαμε ότι μετά από κάθε τελεία στο αρχείο txt θα υπάρχει κενό.

- **int getTotalWords():** Η μέθοδος αυτή αφορά τις συνολικές λέξεις του txt αρχείου. Αθροίζει αναδρομικά τις συχνότητες εμφάνισης της κάθε λέξης, δηλαδή του κάθε κόμβου του δέντρου.
- **int getDistinctWords():** Η μέθοδος αυτή αφορά τον αριθμό των διαφορετικών λέξεων του txt αρχείου. Επιστρέφει μεταβλητή μετρητή (numItems) των κόμβων του δέντρου που ενημερώνεται σε κάθε insert και remove (τρέχει σε χρόνο $O(1)$).
- **int getFrequency(String w):** Η μέθοδος αυτή επιστρέφει τον αριθμό εμφανίσεων στο αρχείο txt της λέξης που δίνεται σαν όρισμα, αλλιώς επιστρέφει NULL αν η λέξη δεν βρίσκεται στο αρχείο.
- **WordFreq getMaximumFrequency():** Η μέθοδος αυτή επιστρέφει την λέξη που εμφανίζεται τις περισσότερες φορές στο txt αρχείο. Η εύρεση της λέξης αυτής γίνεται με αναδρομικές συγκρίσεις των στοιχείων. Επίσης αν το δέντρο είναι κενό, επιστρέφει NULL.
- **double getMeanFrequency():** Η μέθοδος αυτή δίνει την μέση συχνότητα, δηλαδή τον μέσο όρο των συχνοτήτων όλων των διαφορετικών λέξεων. Αυτό προκύπτει αν διαιρέσουμε τις συνολικές λέξεις (με την αναδρομική μέθοδο που χρησιμοποιήσαμε στην getTotalWords) με τον αριθμό των διαφορετικών λέξεων (μεταβλητή μετρητής numItems).
- **void addStopWord(String w):** Η μέθοδος αυτή προσθέτει στην λίστα των stopWords την λέξη που δίνεται σαν όρισμα, μέσω της μεθόδου insertAtFront που ορίσαμε στην κλάση List.java(τρέχει σε χρόνο $O(1)$).
- **void removeStopWord(String w):** Η μέθοδος αυτή αφαιρεί από την λίστα των stopWords την λέξη που δίνουμε σαν όρισμα, μέσω της μεθόδου remove που ορίσαμε στην κλάση List.java .
- **printTreeAlphabetically(PrintStream stream):** Η μέθοδος αυτή εκτυπώνει αλφαβητικά ποιες είναι οι λέξεις του δέντρου και πόσες φορές εμφανίζεται η κάθε μία. Η μέθοδος λειτουργεί αναδρομικά και για κάθε κόμβο καλεί την μέθοδο toString που ορίζουμε στην κλάση WordFreq.java, ώστε οι πληροφορίες να εμφανίζονται όπως θέλουμε.

- **printTreeByFrequency(PrintStream stream):** Η μέθοδος αυτή εκτυπώνει σε αύξουσα σειρά με βάση την συχνότητα τις λέξεις του δέντρου. Η αύξουσα ταξινόμηση υλοποιείται μέσω της quicksort, η οποία επιτυγχάνεται με μέθοδο διάσχισης του δέντρου που πραγματοποιεί προσωρινή αντιστοίχιση των στοιχείων του δέντρου σε πίνακα.

