Κολοβος Κωνσταντινος 3170248
Σταγάκης Βασίλειος 3150164

Δομές Δεδομένων 3η Εργασία

void insert(String w);
αναζητεί αναδρομικά στο δέντρο(O(logN)) για κόμβο που περιέχει το w , εαν βρει αυξάνει συχνότητα αλλιώς προσθέτει κόμβο στο κατάλληλο σημείο μετά απο συγκρίσεις με τα κλειδιά των ήδη υπάρχοντων κόμβων στο δέντρο.

WordFreq search(String w);
Με τη βοήθεια της searchR πραγματοποιεί ανάζητηση αναδρομικά(Olog(N)) και επιστρέφει τον κόμβο με κλειδί w εάν τον βρει.

void remove(String w)
Όπως περιγράφεται στις διαφάνειες εάν βρει κόμβο(O(log(N) αναδρομική αναζήτηση) με κλειδί w τον διαγράφει και επαναφέρει την “ισορροπία” στο δέντρο με τη βοήθεια πεισρτοφών.

void load(String filename);
Μέσω την Bufferedreader διαβάζουμε ένα αρχείο γραμμη-γραμμή , απομονώνουμε τις λέξεις όπως περιγράφεται στις οδηγίες και τις κάνουμε insert στο δέντρο.(O(N))

int getTotalWords();
O(1) μέσω μεταβλητής που ενημερώνεται κατάλληλα.

int getDistinctWords();
O(1) μέσω μεταβλητής που ενημερώνεται κατάλληλα.

int getFrequency(String w);
O(log(N)) χρησιμοποιεί αναζήτηση για να βρει τον κόμβο με κλειδί w.

WordFreq getMaximumFrequency();
O(1) μέσω μεταβλητής που ενημερώνεται κατάλληλα.

double getMeanFrequency();
Με τη βοήθεια της sumFrequencies παίρνει αναδρομικά όλες τις συχνότητες του δέντρου και τις διαιρέι με το πλήθος τους.O(log(N))

void addStopWord(String w);
O(N) Έλεγχος για ύπαρξη στη λίστα του w και εισαγωγή. Στην λίστα δεν δόθηκε η δέουσα σημασία λόγω χρόνου για αυτό έχουμε θέμα με την εργασία. Χρησιμοποιεί τη μέθοδο custom_is_inside της λίστας.

void removeStopWord(String w);
O(N) Έλεγχος για ύπαρξη στη λίστα του w και διαγραφή. Στην λίστα δεν δόθηκε η δέουσα σημασία λόγω χρόνου για αυτό έχουμε θέμα με την εργασία. Χρησιμοποιεί τη μέθοδο custom_remove της λίστας.

void printTreeAlphabetically(PrintStream stream);

Τυπώνει το δέντρο από αριστερά προς τα δεξιά εκμεταλλευόμενη τη φύση του. Αυτό είναι δυνατό γιατι δεν αλλάξαμε την search όπως περιγράφεται στην εκφώνηση(τότε θα ήθελα sort με βάση την αλφαβητική σειρά των κλειδιών). O(log(N))

void printTreeByFrequency(PrintStream stream);

Χρησιμοποιεί αρκετές βοηθιτικές συναρτήσεις αρχικα βάζει όλα τα αντικείμενα του δέντρου σε μια λίστα , μετά βάζει τη λίστα σε array για να μπορέσει να εφαρμοστεί η QuickSort την οποία δεν προλάβαμε να την γράψουμε εξολοκλήρου εμείς λόγω χρόνου και τη βρήκαμε έτοιμη στα ίντερνετς (~~ντροπή μας παρακαλούμε δείξτε έλεος~~) με βάση τη συχνότητα κάθε αντικειμένου , ώστε τελικά να τυπώσει το ταξινομημένο array.O(N)