**Πρώτο μέρος για το μάθημα Ανάπτυξη λογισμικού για πληροφοριακά Συστήματα.**

**Ονόματα Ομάδας**

**Νικολετόπουλος Παναγιώτης 1115201200126**

**Αντώνης Βιρβίλης 1115201200014**

**Παραδοτέα:**

-**main.c** : Περιέχει τις κλήσεις των συναρτήσεων για να αρχίσει το πρόγραμμα

-**functions.**c: Περιέχει τις συναρτήσεις για διάβασμα από αρχείο και σπάσιμο σε λέξεις, εισαγωγή σε trie, διαγραφή από trie, ψάξιμο κειμένου με βάση το trie και όλες τις συναρτήσεις που χρησιμοποιούν .

-**functions.h**: Ο ορισμός των προτύπων των συναρτήσεων και ο ορισμός των δομών trie\_node,paths καθώς και κάποια macros

-**stack.c:** Περιέχει τις συναρτήσεις με τις οποίες βάζω , βγάζω στοιχειά από την δομή stack , την αρχικοποιούν και την διαγράφουν. Πολλές φορές χρησιμοποιώ την δομή stack χωρίς pop και push.

-**stack.h**: Περιέχει τα πρότυπα των συναρτήσεων και τον ορισμό της δομής stack καθώς και μερικα macros

-**test\_main**.c: Η main που χρησιμοποιoυμε για το για το unit testing

-**test.c** :Οι συναρτήσεις για unit testing

-**test.h:** o ορισμός των συναρτήσεων για unit testing

-**makefile**: Η εντολές για το compilation του προγράμματος

Με make project γίνετε compile το πρόγραμμα

Make test\_project γίνετε compile το πρόγραμμα χρησιμοποιώντας την main για unit test

Make run τρέχει το πρόγραμμα με το μικρό dataset

Make pipe τρέχει το πρόγραμμα με το μικρό dataset και κάνει ανακατεύθυνση την έξοδο στο results.txt ώστε να συγκρίνουμε τα αποτελέσματα μετά

Make testrun τρέχει το πρόγραμμα με valgrind για να βρεθούν τυχόν memory leaks

-**το dataset** που μας δώσατε

-**Και μερικές συναρτήσεις** που μας βοήθησαν στο debugging

Αποτελέσματα : To πρόγραμμα για το μικρό dataset τρέχει σε 1.0925 δευτερόλεπτα και τα αποτελέσματα είναι ιδία με τα αποτελέσματα που μας δώσατε (το τσεκάραμε με ένα δικό μας πρόγραμμα και μετά με diff checker online όπου αλλάξαμε μόνο το format της εξόδου μας για να είναι όμοιο)

Από άποψη μνήμης από το valgrind παίρνουμε : All heap blocks were freed – no leaks are possible

Οι μετρήσεις έγιναν από την αρχή πριν αρχικοποιήσουμε το trie και αφού καταστρέψουμε όλο το trie.

Σχεδιαστικές επιλογές:

Η εισαγωγή σε ngram γίνετε αναδρομικά , καθώς και η διαγραφή ngram από trie . O λόγος είναι για ευκολία κώδικα ,αποφυγή και άλλου stack και όπως αναφέρθηκε στο μάθημα το heap από την κλήση των συναρτήσεων δεν μπορεί να πάει πολύ μεγάλο. Παρόλα αυτά θα φτιάξουμε τις αντίστοιχες συναρτήσεις και θα συγκρίνουμε τα αποτελέσματα.

Το ψάξιμο για ngram σε κείμενο γίνετε επαναληπτικά . Ο λόγος είναι ότι εκεί θα γίνονται πολλές επαναλήψεις και πιστεύουμε ότι η συνεχή κλήση των συναρτήσεων αναδρομικά θα φέρει καθυστέρηση .

Για την μη επανάληψη ngrams που έχουν ήδη βρεθεί σε ένα κείμενο χρησιμοποιούμε μια δομή paths όπου κρατεί το μονοπάτι που έχει κάθε ngram στο trie . To paths είναι ένας δισδιάστατος πίνακας . αν γεμίσει τότε διπλασιάζετε . Όταν βρούμε μια λέξη στο κείμενο πρώτα τσεκάρουμε αν υπάρχει το αντίστοιχο μονοπάτι στο paths και μετά το εμφανίζουμε. Η αναζήτηση στο paths είναι σειριακή . Έγινε προσπάθεια να γίνει δυαδική αναζήτηση αλλά θα έπρεπε πάλι να πληρώσουμε τις μετακινήσεις των paths αφού πρέπει κάθε φορά να είναι sorted. H χρήση της memove δεν βοήθα σε αυτή την περίπτωση. Προς το παρόν έχουμε μια απλοποιημένη μορφή της paths και εμφανίζουμε τα αποτελέσματα της αναζήτησης όπως τα βρίσκουμε και όχι μόλις φτάνουμε στο F . Έχουμε βρει λύση και στην εμφάνιση ανά ριπές και στην αποτελεσματική αναζήτηση. Η υλοποίηση της θα γίνει στο δεύτερο κομμάτι.

Προς το παρόν έχουμε ένα standar μέγεθος για τις λέξεις . Επομένως για ένα παράλογο μέγεθος λέξης στο trie δεν θα χωρέσει.

Για το κομμάτι του unit testing :

Για το unit test της add και της delete τσεκάρουμε αν μετά την add και την delete του ngram υπάρχει το ngram στο trie ,Μετά την add περιμένουμε να υπάρχει και μετά την delete περιμένουμε να μην υπάρχει η να γυρνά ότι δεν μπορεί να την σβήσει , η ότι άλλαξε τον τελικό κόμβο σε μη τελικό .

Για την συνάρτηση αναζήτησης ελέγχουμε τις συναρτήσεις τις δυαδικής αναζήτησης με μερικά test cases και αν υπάρχει το μονόπατη στο paths.