Θωμάς Μαλιάππης 1115201400093
Γιώργος Αποστόλου 1115201400015

Project Part 1

main.cpp

Διαβάζουμε και ελέγχουμε τα ορίσματα. Στην συνέχεια καλούμε την trieCreate ώστε να δημιουργήσουμε το trie με τα Ngrams απο το αρχείο initFile που δίνεται ως όρισμα. Επίσης καλούμε την queryRead που διαβάζει queries απο το αρχείο queryFile που δίνεται ως όρισμα και κάνει τις κατάλληλες εκτυπώσεις.

trie.h

```
class TrieNode {        // κόμβος trie
   int capacity;              // μέγεθος πίνακα παιδιών
   int childNum;              // αριθμός “active” παιδιών
   bool final;                // αν ο κόμβος είναι τελευταία λέξη Ngram
   char * word;               // λέξη του κόμβου
   TrieNode **childs;         // δείκτης στον πίνακα των παιδιών του κόμβου
};

class Head {            // κόμβος head του trie
   TrieNode * root;           //δείκτης στον πρώτο κόμβο του trie
};
```

Επίσης υπάρχουν δηλώσεις συναρτήσεων.

trie.cpp

free_trie: Αναδρομική διαγραφή του trie

trieCreate: Ανοίγουμε το αρχείο initFile από όπου διαβάζουμε Ngrams και για κάθε Ngram το αποθηκεύουμε σε έναν πίνακα και καλούμε την insertNgram για να το εισάγουμε στο trie.

insertNgram: Για κάθε λέξη του Ngram που εισάγουμε ελέγχουμε αν υπάρχει κόμβος. Αν υπάρχει, πηγαίνουμε στα παιδία του και ψάχνουμε αν υπάρχει κόμβος για την επόμενη λέξη του Ngram. Αν δεν υπάρχει , τότε φτιάχνουμε καινούργιο κόμβο-παιδί και στην περίπτωση που δεν υπάρχει χώρος στον πίνακα γίνεται διπλασιασμός του.

searchNgram: Για κάθε λέξη του Ngram που ψάχνουμε ελέγχουμε αν υπάρχει κόμβος. Αν υπάρχει συνεχίζουμε να ψάχνουμε στα παιδιά του την επόμενη λέξη αλλιώς σταματάει η αναζήτηση. Η αναζήτηση γίνεται με binary-search.

deleteNgram: Ψάχνουμε να βρούμε αν υπάρχει το Ngram που θέλουμε να διαγράψουμε. Αποθηκεύουμε σε έναν πίνακα το μονοπάτι των κόμβων που έχουμε ακολουθήσει. Για κάθε κόμβο του path ελέγχουμε αν ο κόμβος του path είναι τελευταία λέξη του Ngram. Αν είναι και δεν έχει παιδιά τότε διαγράφεται, αλλιώς αλλάζει το final. Αν δεν είναι και δεν έχει παιδιά διαγράφεται αλλιώς δεν γίνεται τίποτα.

queryRead: Ανοίγουμε το αρχείο queryFile από όπου διαβάζουμε queries και για κάθε query το αποθηκεύουμε σε έναν πίνακα και καλούμε την κατάλληλη συνάρτηση ανάλογα με την πρώτη λέξη του query.

sortChilds: Βρίσκουμε την θέση που πρέπει να μπεί ο καινούργιος κόμβος και μετακινούμε όσους κόμβους βρίσκονται από εκεί και δεξία κατα μία θέση δεξιά.

List.cpp

Χρησιμοποιούμε την δομή της λίστας χωρίς διπλότυπα για να εισάγουμε τις σωστές απαντήσεις της Q που πρέπει να έχουμε συγκεκριμένη σειρά και να μην εμφανίζονται περισσότερες απο μία φορές.