Προγραμματισμός Συστήματος

Project 1

Ονοματεπώνυμο: Βασιλική Γεωργάρα

A.M.: 1115201400026

Πώς δουλεύει το πρόγραμμα:

Αρχικά διαβάζει τα ορίσματα, είτε 5, είτε 3 (ανάλογα αν θα δοθεί k από τον χρήστη ή θα κρατηθεί το default που είναι 10).

Γίνεται έλεγχος για κάθε όρισμα και φυσικά μπορούν να δοθούν και με αντίθετη σειρά.

Στη συνέχεια δημιουργείται το Trie, το οποίο δημιουργεί την ChildList, PostingList και Tnode. Το Trie αρχικοποιείται δεσμεύοντας τη ρίζα, αρχικοποιώντας το Tnode της ρίζας και δημιουργώντας την ChildList, PostingList με prev=NULL και value = ' '.

Επίσης, δημιουργείται και το Μαρ, όπου δημιουργεί δείκτες Document, start και end και τελικά αρχικοποιείται με count=totalwords=0 και start=end=NULL.

Ύστερα, ανοίγω το file μέσα σε συνάρτηση, περνώντας το trie και το map: ανοίγω το file και το διαβάζω μέχρι να τελειώσει, καθαρίζοντας τις γραμμές της μεταβλητής line από σκουπίδια βάζοντας σε όλα 0. Έπειτα, καλείται η parseLineAndAddDocument μέσα στην οποία αποθηκεύεται το id από το κείμενο και γίνεται ο έλεγχος για το αν δόθηκαν σε σειρά και τερματίζει το πρόγραμμα σε αντίθετη περίπτωση.

Δημιουργώ τη δομή Document, περνώντας το id και δημιουργώντας τη λίστα λέξεων.

Για κάθε λέξη του document, βάζω τη λέξη στη λίστα λέξεων του document, καθώς και την προσθέτω στο trie. Για να την προσθέσω στο Trie, παίρνω έναν-έναν χαρακτήρα και αν υπάρχει ήδη στο trie, τότε επιστρέφω τον κόμβο για να συνεχίσω στον επόμενο χαρακτήρα, αλλιώς δημιουργώ τον κόμβο και τον προσθέτω. Όταν τελειώσω με τη λέξη,

δημιουργώ την PostingList και την ενημερώνω, ψάχνοντας αν η λέξη εισάγεται 1η φορά ή όχι, ώστε να ενημερώσω κατάλληλα και τη μεταβλητή frequency.

Για τις ανάγκες του score, υπάρχει η μεταβλητή totalwords που ενημερώνεται με κάθε πλήθος document words. Όταν τελειώσει η διαδικασία για το 1ο document τότε προστίθεται στο map. Τέλος, ξεκινάει η διαδικασία των εντολών. Πάντα θα φαίνεται η γραμμή εντολών "Minisearch>" όπου θα αναμένεται η πληκτρολόγηση των εντολών με "/" πριν από κάθε μία.

Αν δοθεί «/exit» θα τερματίσει η επανάληψη για το διάβασμα των εντολών και θα ακολουθήσουν τα απαραίτητα destructions.

Αν δοθεί σκέτο «/df» τότε θα καλεστεί η df function που καλεί τη συνάρτηση document_frequency, η οποία αν ο τρέχον κόμβος του Trie υπάρχει τότε καλεί την getWord για να επιστρέψει τη λέξη και την τυπώνει και ύστερα, ΑΝΑΔΡΟΜΙΚΑ, καλούνται όλοι οι κόμβοι παιδιά, ώστε τελικά να εκτυπωθούν οι λέξεις με τις συχνότητές τους.

Επίσης, έχει υλοποιηθεί και μια συνάρτηση strrev η οποία καλείται στην getWord ώστε να αντιστρέψει τη λέξη που διασχίζεται στο δέντρο με τη σωστή σειρά. Έτσι, τελειώνει ο έλεγχος για τις μονολεκτικές εντολές.

Όσον αφορά την «/df word» καλείται η df_word function που ψάχνει τη λέξη στο Trie.

Η «/tf id word» αποθηκεύει το id από την εντολή και τη λέξη και καλεί τη συνάρτηση tf η οποία κάνει 2 αναζητήσεις: πρώτα ψάχνει τη λέξη στο Trie και, αν δεν τη βρει, επιστρέφει 0, αν τη βρει όμως, κάνει αναζήτηση και στην PostingList το freq του συγκεκριμένου id.

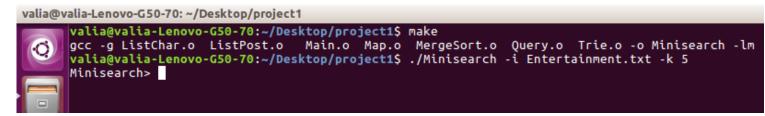
Τέλος, αν έρθει η εντολή «/search query1 query2 ... queryn», από τις η queries που θα πληκτρολογήσει ο χρήστης, θα διαβαστούν μόνο οι πρώτες 10 και θα αποθηκευτούν σε μια λίστα με queries όπου θα φτιαχτεί και ένας πίνακας για τα scores που θα έχει μέγεθος όσα και τα documents. Ύστερα, καλείται η prepareQuery όπου υπολογίζεται το score, δηλαδή υπολογίζεται ο κάθε όρος και στέλνεται σε κατάλληλη συνάρτηση για να υπολογιστεί το τελικό κλάσμα. Και η printQuery που

εκτυπώνει τα k αποτελέσματα. Η εκτύπωση των αποτελεσμάτων των queries χρησιμοποιεί και την print από το αρχείο ListChar, όπου οι λέξεις που συμπεριλαμβάνονται στη search εντολή και υπάρχουν στα κείμενα, βρίσκονται μεταξύ των «|» για τις λέξεις που έχουν την μεταβλητή marked=true. Για κάθε νέα αναζήτηση όλα «καθαρίζουν» και όσες λέξεις ήταν μαρκαρισμένες, πλέον θα περάσουν από τη συνάρτηση unmark για να εκτελεστεί επιτυχώς η νέα αναζήτηση. Για την mergeSort που χρησιμοποιείται, ο κώδικας πάρθηκε από την ιστοσελίδα https://www.geeksforgeeks.org/merge-sort/.

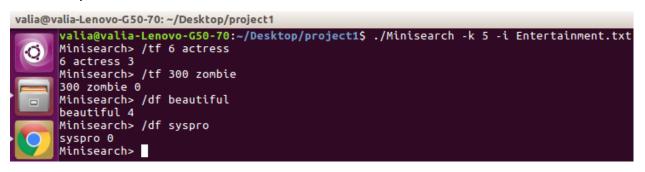
Ακολουθούν ενδεικτικές εκτελέσεις παρακάτω.

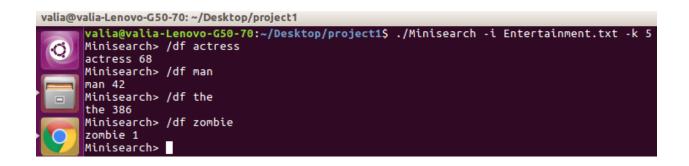
Ενδεικτικές εκτελέσεις του προγράμματος:

Η μεταγλώττιση του προγράμματος γίνεται με το «make» και είναι separate compilation, υλοποίηση που επιταχύνει την μεταγλώττιση.



Ορισμένες εκτελέσεις για τις εντολές /df word και /tf id word για το αρχείο Entertainment.txt.





Αν δοθεί λάθος αρχείο, πχ λόγω τυπογραφικού λάθους:



Επίσης, αν δοθούν λάθος αριθμημένα αρχεία:

```
valia@valia-Lenovo-G50-70: ~/Desktop/project1

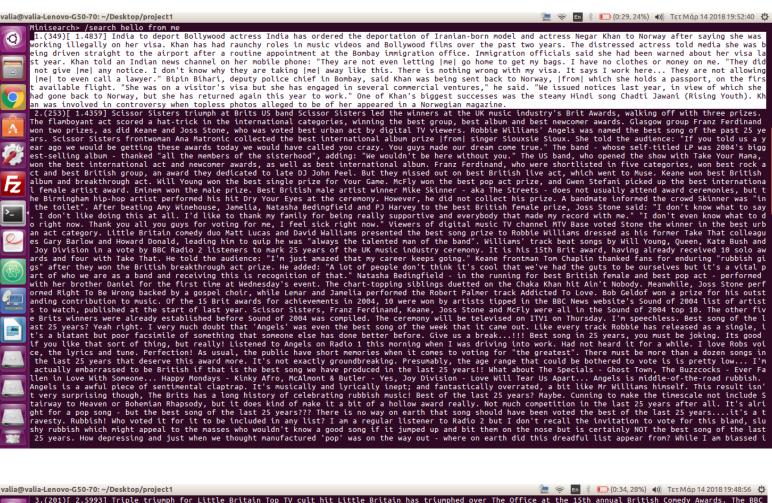
valia@valia-Lenovo-G50-70: ~/Desktop/project1$ ./Minisearch -k 5 -i Sport.txt

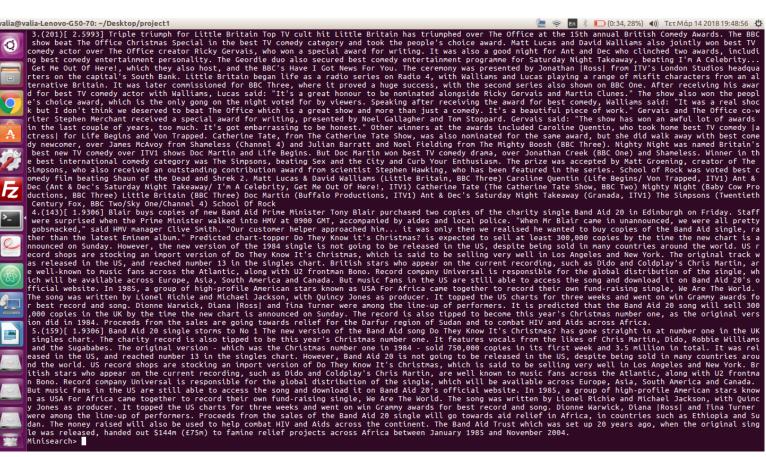
Documents are not ordered. Exiting program...valia@valia-Lenovo-G50-70: ~/Desktop/project1$
```

Μια ενδεικτική εκτέλεση του df:



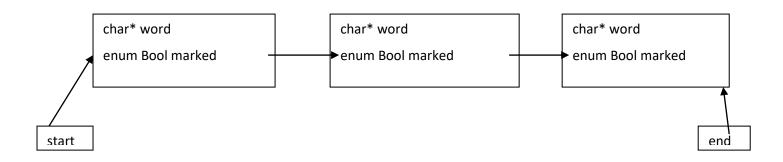
Μια ενδεικτική εκτέλεση του search:



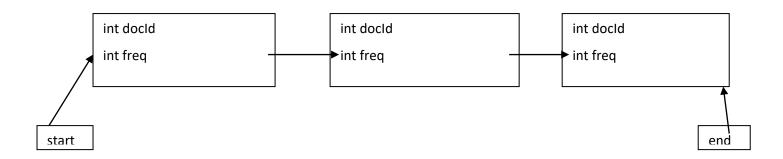


Οι δομές του προγράμματος:

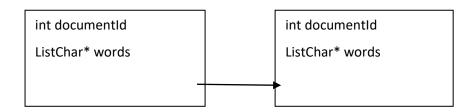
ListChar: (λίστα λέξεων)



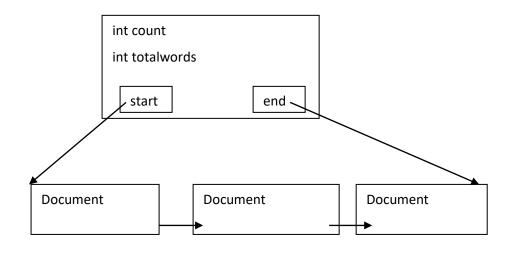
ListPost: (posting list)



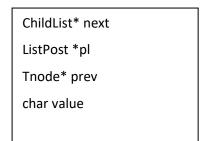
Document:



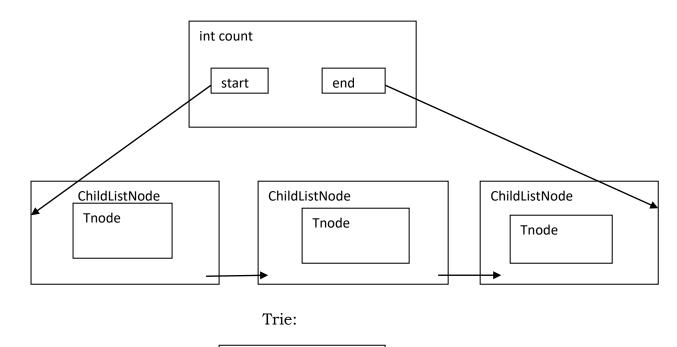
Map



Tnode:



ChildList:



Query:

Tnode *root

