

ΙΩANNA ΛΥΚΟΥΔΗ  
201400091

**Compilation:** make

**Τρόπος εκτέλεσης:** ./minisearch -i fullDataset/smallDataset.txt -k 2 (παράδειγμα)

Map.h: Ορίζω μία δομή map η οποία αποτελείται από δύο int ( id & wamount) καθώς και έναν δείκτη σε char. Η συγκεκριμένη δομή χρησιμεύει στην πρώτη αποθήκευση του αρχείου που διαβάζω. Id του εκάστοτε κειμένου, wamount το πλήθος των λέξεων κάθε κειμένου και line για να εκτυπώσω το κείμενο μου στην πορεία .

Parcer.h: Εδώ αναφέρεται η συνάρτηση την οποία ορίζουμε στο .c

Parcer.c: Ορίζω την συνάρτηση με την οποία διαβάζω τις μεταβλητές που χρειάζομαι από την γραμμή εντολών, δηλαδή το Document καθώς και τον αριθμό που θέλω για να εκτυπώσω τα top-k μεγαλύτερα score.

Trie.h: Ορίζω δύο δομές. Η πρώτη είναι για την λίστα που θα χρειαστώ για το Posting List. Η δομή ορίζεται από το id του εκάστοτε κειμένου, την frequency της λέξης μέσα στο κείμενο και έναν δείκτη στο επόμενο στοιχείο της λίστας. Η δεύτερη είναι για το Trie. Η δομή αποτελείται από τον χαρακτήρα που υπάρχει στον κόμβο και δείκτες σε child, next & Posting List. Αναφέρονται επίσης οι συναρτήσεις που χρησιμοποιούνται στο trie.c

Trie.c: Αρχικά ορίζω συναρτήσεις για να μεταχειρίζομαι την λίστα μου. Η createListNode δημιουργεί έναν κόμβο και αρχικοποιεί τις τιμές του. Η insertList εισάγει δεδομένα στη λίστα, ενώ παράλληλα ελέγχει περιπτώσεις, όπως να υπάρχει ήδη κόμβος για το συγκεκριμένο id που την κάλεσα ή να μην υπάρχει καθόλου. Η cleanList απελευθερώνει τον χώρο που έχω δεσμεύσει για την λίστα. Η createTrieNode δημιουργεί κόμβο για το Trie μου και αρχικοποιεί τις τιμές του. Η insertTrie εισάγει στο Trie μία συγκεκριμένη λέξη, με συγκεκριμένο id. Στην περίπτωση που υπάρχουν ήδη λέξεις σε αυτό πρέπει να της ταξινομεί αλφαβητικά οπότε και κάνει αντίστοιχους ελέγχους. Η searchTrie αναζητά στο Trie μια λέξη και επιστρέφει τον κόμβο με το Posting List της. Η cleanTrie απελευθερώνει τον χώρο που έχω δεσμεύσει για την λίστα.

Heap.h: Ορίζω δύο δομές σωρού. Η πρώτη περιλαμβάνει δείκτη σε right, left, parent, score & id και αντιστοιχεί σε ένα κόμβο. Η δεύτερη δομή Heap μας χρησιμεύει για να γνωρίζουμε πόσους κόμβους έχουμε σε όλη την υλοποίηση.

Heap.c: Εδώ ορίζω τις συναρτήσεις που χρειάζομαι για να διαχειρίζομαι το heap μου. Η CreateHeapNode δημιουργεί έναν κόμβο και αρχικοποιεί τα δεδομένα του. Η CreateHeap κάνει επίσης το ίδιο. Η heapify είναι υπεύθυνη για την ταξινόμηση του heap, αφού θέλουμε η ρίζα του heap να έχει το μεγαλύτερο score. Η InsertHeap εισάγει αριθμό (το score & το id) στο heap και

στο τέλος το ταξινομεί κιολας. Η cleanHeapNode & cleanHeap απελευθερώνουν ότι χώρο έχουν δεσμεύσει.

Minisearch.c: Καλώ την read\_args για να αποθηκεύσω τους αριθμούς που θέλω από την γραμμή εντολών. Ανοίγω το Αρχείο. Δεσμεύω χώρο σε μια μεταβλητή buffer για να μπορώ να διαβάζω μέσω της getline. Φτιάχνω ένα array (τύπου map) για να αποθηκεύσω εκεί τα id των κειμένων καθώς και τα “documents”. Θα μετρήσω όλες τις γραμμές του αρχείου μου για να του δεσμεύσω τόσο χώρο. Παίρνω μία μία γραμμή και αφού χρησιμοποιήσω την strtok παίρνω τα δεδομένα μου ( id, document). Κάνω έναν ελέγχο: σε περίπτωση που το αρχείο μου δεν έχει σωστή δομή (δεν πήρα id πχ με την πρώτη εφαρμογή της strtok) τότε ενημερώνω. Χρησιμοποιώ και ένα flag (=wrong) για όταν οι γραμμές είναι 0, αλλά και για την παραπάνω περίπτωση με το id. Επίσης, ελέγχω την ακολουθία των id (πρέπει να είναι αύξουσα). Δημιουργώ τον αρχικό κόμβο για το trie μου. Για όσο οι γραμμές μου δεν έχουν τελειώσει: διαχωρίζω τις λέξεις τους και τις εισάγω μία μία το Trie. Επίσης μετράω πόσες λέξεις εισάγω γιατί χρειάζεται στον υπολογισμό του score. Για να διαβάζει από το stdin χρησιμοποιώ την getline. Αν διαβάσω \exit τερματίζω. Για \search καλώ την αντίστοιχη συνάρτηση για το trie, αυτή που επιστρέφει τον κόμβο που έχω το Posting List της λέξης και έτσι μπορώ και εκτυπώνω ότι θέλω. Εδώ γίνεται και ο υπολογισμός του score. Εισάγω στο heap το score & id. Για την \df έχω δύο συναρτήσεις που είτε προσπελαίνουν όλο το δέντρο, είτε απλά εκτυπώνουν τις πληροφορίες για μία συγκεκριμένη λέξη. Για την \tf αποθηκεύω το id & την λέξη. Ψάχνω την λέξη στο δέντρο και εκτυπώνω το frequency για το συγκεκριμένο id. Οτιδήποτε άλλο πληκτρολογηθεί δεν αναγνωρίζεται. Απελευθερώνω ότι έχω δεσμεύσει.