

README

Όνομα : Κωνσταντίνος
Επώνυμο: Χρήστου
ΑΜ: 1115201400229

COMPILE

Για compile τρέχουμε την εντολή 'make'

RUN

`./stucks -b <<hashtable number>> -l <<input file>>`

Για παράδειγμα:

`./stucks -b 20 -l 300+.in`

IMPLEMENTATION NOTES

class structure: Δημιουργείται object της class structure, το οποίο περιέχει και την skiplist και το hashtable. Στον constructor της class structure δημιουργείται object της skiplist και object του hashtable. Από την class structure καλούνται οι συναρτήσεις της skiplist και του hashtable.

skipList: Αρχικά δημιουργούνται ο αρχικός κόμβος (header) με το maximum level, ο οποίος δεν παίρνει δεδομένα record και ο τελικός κόμβος με maximum level και maximum value ο οποίος επίσης δεν παίρνει δεδομένα record. Η εισαγωγή στοιχείων στην skiplist γίνεται ταξινομημένα. Η επιλογή για το επίπεδο του κάθε κόμβου γίνεται τυχαία. Στην συνάρτηση `percentiles2` κρατάω 2 πίνακες που αντιστοιχίζω postcode με counter, για να ξέρω στο κάθε postcode πόσα στοιχεία υπάρχουν για να μπορέσω να υπολογίσω τα ποσοστά. Στην συνάρτηση `bottom2` δημιουργώ έναν κλώνο-λίστα της skiplist, την οποία διατρέχω στην συνάρτηση `bottom3` για να βρω τους ζητούμενους k χαμηλότερους μέσους όρους.

hashtable: Δημιουργείται ένας πίνακας δεικτών σε λίστες με τον constructor του hashtable. Κάθε θέση του πίνακα αντιστοιχίζεται σε ένα postcode και περιέχει μία λίστα από στοιχεία που έχουν το ίδιο postcode. Στην συνάρτηση `taverage2` δημιουργείται ένας κλώνος-λίστα της λίστας της θέσης του hashtable που αντιστοιχεί σε ένα ζητούμενο postcode, και στην συνάρτηση `taverage3` διατρέχεται αυτή η λίστα προκειμένου να βρεθούν οι k καλύτεροι μέσοι όροι των μαθητών στο συγκεκριμένο postcode. Η συνάρτηση `hash` είναι από τον σύνδεσμο που δίνεται στην εκφώνηση για επεξήγηση του chain hashing. Η υλοποίηση της `delete_node2` γίνεται με βάση μια υλοποίηση που βρήκα στο διαδίκτυο.