ΑΜ:1115201800009 ΤΥΠΑΛΔΟΣ-ΠΑΥΛΟΣ ΑΠΟΣΤΟΛΑΤΟΣ

disclaimer: αυτη η εργασια παει χερι χερι με την τελευταια του μαθηματος δομες δεδομενων(K08) οπου υλοποιησαμε ολες τις δομες που ζηταει αυτη η εργασια και επισης μας δοθηκε το template και η νοοτροπια για τα makefiles. βεβαια ο κωδικας φρεσκαριστηκε για τις ιδιατεροτητες της παρουσας εργασιας
επελεξα C++ γιατι η εκφωνηση μου φαινεται πιο πολυ αντικειμενοστραφη παρα διαδικαστικη και επισης το porting του hashtable απο c σε c++ αλλα και λειτουργιες οπως πρακτικη γνωση σε file handling moy φαινονται επικοδομητικα ως προς την μαθηση μου και συλλογη κωδικα

περιγραφη αντικειμενων και ροης προγραμματος:
το αντικειμενο hash αποθηκευει το μεγεθος του πινακα και το inverted idex μιας και το ενα δεν υφισταται χωρις το αλλο και θα ηταν επωδυνο να περνιουνται ως ορισμα σε καθε συναρτηση. απεφυγα να γραψω καινουρια classes και συναρτησεις λιστων για καθε data type(student,int κτλ) για αυτο οι λιστες του πινακα και του inverted idex χρησιμοποιουν την ιδια κλαση με pointer σε student με την παραδοχη οτι στο inverted idex τα χειριζομαι σαν να ηταν references. Για καθε λιστα μαθητων το inverted index αποθηκευει το ετος τους και ποσοι μαθητες περιεχει η καθε λιστα(scount).

το config file θετει το μεγεθος του πινακα και το μεγιστο πληθος χαρακτηρων των χαρακτηριστικων των φοιτητων για να μην γινει segmentation fault στο standard input της c
η main δινει δefault τιμες σε αυτες τις τιμες και ψαχνει το command line για ορισματα.αν βρει config file τα αλλάζει,αν δεν βρει config αλλα inputfile κανει την προτεινομενη απο τους διορθωτες ευρεση γραμμων του αρχειου και θετει το size αναλογα(πιστευω 150% των γραμμων ειναι λογικο συνεχεια>*) και μετα προσθετει τις πλειαδες φοιτητων του αρχειου στο hashtable και επειτα αν δεν υπαρχει ηδη κανει sorted insert στο inverted index (δηλαδη στη σειρα του II πρωτα ειναι η λιστα του 2010 μετα 2011 κτλ). Λογω του hashcode και το γεγονος οτι γινετα insert στην αρχη της λιστας εχει O(1) πολυπλοκοτητα.

* Στις εντολες του prompt προσθεσα μια s(how) που εκτυπωνει ολο το hashtable και ετσι μπορει κανει να δει ποσα colissions υπαρχουν

Το ιδιο και στην αναζητηση με την προυποθεση οτι εγω και ο χρηστης εχουμε παρει σωστες επιλογες για την αποφυγη collisions.
Στην διαγραφη φοιτητη τον βγαζει απ’το hashtable και επιτρεφει τον pointer του και το ετος του με references για να μπορω να ξερω που βρισκεται στο II για να μπορω να τον βγαλω και απο κει και διαγραφω τον pointer του μετα για να μπορω να συκρινω το id του χωρις memory error

οι συναρτησεις του prompt που σχετιζονται με ετη χρησιμοποιουν την συναρτηση getNthyear(y) Που επιστρεφει τον κομβο inverted index που αντιστοιχει στο ετος y και το scount.

Οι count,number year,minimum και average νομιζω ειναι αυτονοητες.
Στην top num year επελεξα να κανω μια quicksort τους μαθητες βασει των βαθμων παρα να βρω τον καλυτερο Num φορες δηλαδη καλυτερα o(nlogn) παρα O(n*num). Ζητειται να εκτυπωνονται και αυτοι που ισοβαθμουν αρα εκτυπωνω μεχρι να συναντησω διαφορετικο βαθμο σε φθινουσα ακολουθια num φορες.

Στην Postal rank πρεπει να προσπελασω το zip καθε φοιτητη. Για καθε τετοιο στην 3η for ψαχνω αν το εχω ξαναβρει. Αν ναι αυξανω κατα 1 τις φορες που το εχω βρει,τα zips ομως ειναι ταξινομημενα κατα φθινουσα σειρα αρα με αυτο το +1 μπορει να ξεπερασε καποια στοιχεια αριστερα του οποτε του κανω

αντιμεταθεση μεχρι να μπουν ολα αυτα δεξια του. Η εκτυπωση ειναι ιδια με το topnumberyear μονο που εδω ζητειται να εκτυπωνω την num(rank) καταταξη οχι τη 1 μεχρι num,αν φτασω στο τελος του πινακα πριν φτασω την καταταξη rank ειναι error `"rank requested is greater than the zip ranks"`

κανω παραδοχη οτι για να μην δημιουργησω καινουρια δομη για μια μονο διεργασια κανω allocate μνημη για το worst case οπου καθε φοιτητης εχει διαφορετικο zip κατι που ειναι υπερβολικα brute force

Στην exit ο destructor του hashtable και του inverted index διαγραφει τους student pointers(οχι ο list destructor γιατι δε θα μπορουσε να δουλεψει η delete του prompt και μετα διαγραφονται τα ιδια. Με valgrind -s και –leak-check=full δεν εχω ουτε leak ουτε error.
Makefile:
g++ για c++ -wall για warnings -g3 για debugging στο gdb. φτιαχνω φακελο ωστε τα .o αρχεια να μπαινουν εκει και να μη μπερδεονται με τα .cpp. Για καθε cpp πρεπει να δηλωσω ποια αλλα αρχεια χρειαζεται για να πετυχει μεταγλωτισση. Θετω σε ολα το classes.hpp γιατι αυτο κανουν ολα include. Μετα καθε cpp μεταγλωτιζεται ξεχωριστα με τις παραπανω εντολες.εχω θεσει εκτελεσιμο το mngstd και τα .OBJ αρχεια ενωνωνται για να το φτιαξουν.

Εκτελεσιμο: ./mngstd
Links που βοηθησαν:
https://www.geeksforgeeks.org/quick-sort/

θεωρω αν καποιος διαβασει το pdf αυτο πριν δει τον κωδικα και συμβουλευεται τους αστερισκους οταν κοιταει τις 10 λειτουργιες θα καταλαβει την ροη και τους αλγοριθμους και τα προγραμματιστικα της C ειναι τετριμμενα.
Υπενθυμιση
list=pointer σε tablentry(πιο abstract ονομασια)
II=pointer σε inverted index node