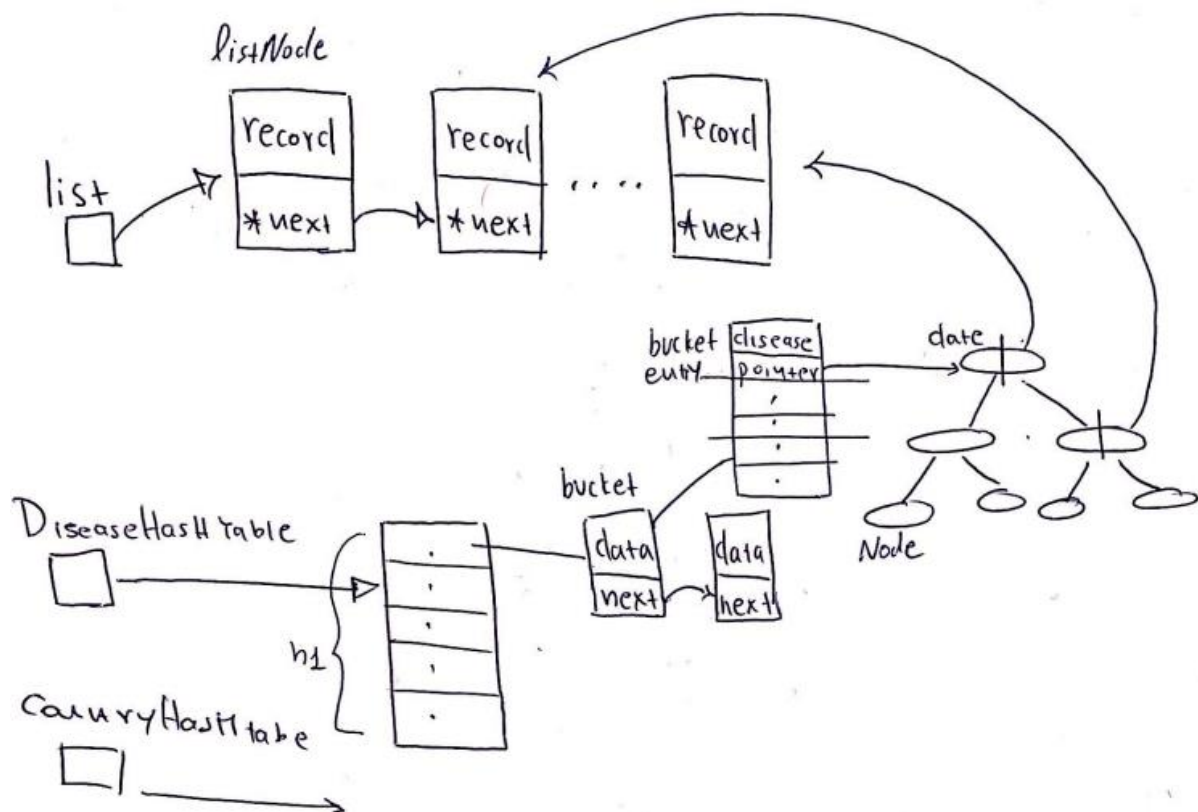


❖ Τρίγκας Ευστάθιος

❖ *sdi1700167*

1. `chmod +x ./validator.sh`
2. `make`
3. `validator.sh small.txt 10 10 1024`



Αρχικά δημιουργώ μια λίστα και βάζω τα δεδομένα από το αρχείο με την `insertRecordsFileToLinkedList()`. Αυτή η συνάρτηση ανοίγει το αρχείο και για κάθε γραμμή του παίρνει τα δεδομένα και τα κάνει copy στον καινούριο κόμβο της λίστας.

Έπειτα δημιουργώ με την συνάρτηση `initializeHashTable()` δυο hashtables. Η συνάρτηση αυτή παίρνει σαν όρισμα έναν άδειο πίνακα από δείκτες και για κάθε θέση του δημιουργεί μια δομή bucket. Η δομή αυτή απαρτίζεται από έναν δείκτη ο οποίος να χρησιμοποιηθεί για να δημιουργηθεί αλυσίδα από bucket σε περίπτωση που γεμίσει και έναν δείκτη σε έναν άλλο πίνακα. Αυτός ο πίνακας ουσιαστικά αναπαριστά τα δεδομένα που βρίσκονται μέσα στο bucket. Ο πίνακας αυτός έχει έναν αριθμό από δομές τύπου bucket entry. Κάθε θέση αυτού του πίνακα με τύπο bucket entry απαρτίζεται από έναν δείκτη σε χαρακτήρα έτσι ώστε να αποθηκεύεται το όνομα της ασθένειας ή της χώρας αλλά και έναν δείκτη σε δένδρο ο οποίος έχει συγκεκριμένη δομή που θα εξηγήσω αργότερα.

Αφου αρχικοποιηθοθν τα hashTables (δεδομενα null) ηρθε η ωρα να εισαγουμε τα δεδομενα μας.

Για αυτό καλείτε η συνάρτηση insertdataToHashTable(). Η συνάρτηση αυτή ανάλογα το dataType Που θα της δοθεί(diseaseId για το diseaseHashTable και country για το countryHashTable) γεμίζει τον αντίστοιχο πίνακα με τα δεδομένα από την λίστα. Αυτό το κάνει με τον εξής τρόπο. Αρχικά παίρνει έναν έναν τους κομβό της λίστας και παίρνει το αντίστοιχο δεδομένο country η diseaseId και το χασαρι. Μετα πηγαίνει στην αντίστοιχη θέση του πίνακα παίρνει τον δείκτη σε bucket πηγαίνει στο bucket μετα στα δεδομενα και εκει ελέγχει αν ο δείκτη της ασθένειας είναι

άδειος. Αν είναι σημαίνει ότι δεν υπάρχει αρα δημιουργείτε ένα καινούργιο δένδρο και ο δείκτης τοθ κλειδιού δείχνει πλέον στο δεδομένο της λίστα. Αν υπάρχει απλα προστίθεται μια ημερομηνια στο δενδρο. Για το δενδρο τωρα. Αρχικά η δομη τοθ απαρτιζεται από μια ημερομηνία εισαγωγής τοθ ασθενούς και αριστερο και δεξι παιδί. Η συναρτηση insert() χρησιμοποιείτε για να δημιουργηθεί καινούριο δενδρο η και να προστεουν δεδομενα σε αυτό. Αν το δένδρο δεν υπάρχει τοτε απλά δημιουργείτε ενασ καινουργιος κομβος και επιστρεφεται.Αν ομως το δενδρο υπάρχει τοτε γίνεται κλασικη εισαγωγή σε δυαδικό δένδρο ανάλογα με τις ημερομηνιες αναδρομικά. Αθο γίνεται με τον εξησ τρόπο. Πηγαινω αναδρομικά στα φυλα του δενδρου σε σωστη ύφεση ανάλογα με την ημερομηνία. Μολις βρω άδεια θέση δημιουργώ έναν κομβο και αμεσος μετρά τσεκάρω αν το δενδρο χρειάζεται να ισορροπηθεί. Αν ναι τοτε χρησιμοποιώ τις 4 περιπτώσεις για να καταλάβω αν χρειάζεται rightrotate η leftrotate.ετσι το δένδρο παραμένει ισορροπημένο.

Τωρα αφου διμιουργηθηκαν οι βασικες δομες αρχιζοθμε να εκτελοθμε εντολεσ πανω στην βαση .

/globaldiseaseStats

Πηγαινει στον diseaseHasTable και για κάθε ασθeneia παίρνει το δενδρο της και μετραει τους κόμβου του. Αν του εχουν δοθει ημερομηνίες μετραει αυτους μεσα στο διαστημα.

Και τα θπολοιπα ερωτηματα απλα χρησιμοποιουν τις ιδι υπαρχουσες δομες για να πάρουν αυτό που ζητείτε.

/topk diseases και /topkCountry

Και οι δυο συναρτησεις υλοποιουνται με παρομοιο τροπο.

Ουσιαστικα διμιουργεите μια στοιβα δενδρου.

Εστω ότι μιλαμε για την topkdiseases.

Αρχικά χρθσιμοποιωντας τα δενδρα που εχουμε για καυε ασθeneia διμιουργοθμε έναν πίνακα ο οποιος εχει ολεσ τις ασθeneies με τον αριθμο αυτόν που εχουν ασθeneησει. Είναι δηλαδι πίνακας με δεδομενα δομης ασυeneia-αριυμος ασθενων.(αν θοθει και country) τοτε προσαρμοζετα αναλογα για την συγκεκριμενη χωρα.αφοτου διμιουργηγει ο πίνακας διμιουργите ένα διαδικο δενδρο το οποιο ο κάθε κομβος του δεχνει σε μια θέση του πίνακα.Μολις διμιουργηγει καλείτε η συναρτηση heapify η οποιο κανει αυτό το δενδρο σωρο max και τελος εμφανιζονται τα K πρωτα δεδομενα αυτου του σωρου