Ιωάννης Δαλιάνης

1115201700027

Project 1

Προσοχή στην εκτέλεση του προγράμματος χρειάζεται να δίνεται το σωστό path για το

αρχείο εγγραφών, πχ "Assets/fakeDiseases.txt".

Η επιλογή make κάνει copile το πρόγραμμα. Η επιλογή make all κάνει compile και εκτελεί το πρόγραμμα με μία από τις σχολιασμένες γραμμές εκτέλεσης.

Τα δεδομένα για τα diseaseRecords των ασθενών αποθηκεύονται μία μόνο φορά στη μνήμη.

Χρησιμοποιείται μια Απλά Συνδεδεμένη Λίστα για να κρατάει τους αρχικούς δείκτες προς

τα diseaseRecords η οποία δε μεταβάλλεται καθόλου μέχρι το τέλος της εκτέλεσης του προγράμματος,

με εξαίρεση την περίπτωση της querry insertPatientRecord. Οι άλλες δομές αποθηκεύουν πάλι

μόνο δείκτες στα diseaseRecords και όταν αυτές σβήνονται, δε πειράζουν καθόλου

τα diseaseRecords. Τα diseaseRecords σβήνονται μόνο στο τέλος που αδειάζει η Απλά

Συνδεδεμένη Λίστα.

Τα recordId αντιμετωπίζονται σα strings και η σύγκρισή τους γίνεται με strcmp, σύμφωνα με απάντηση

που δόθηκε στο Piazza.

Για όλες τις δομές έχουν υλοποιηθεί συναρτήσεις εκτύπωσης.

Όταν φτιάχνεται ένα καινούριο patientRecord από αρχείο ή με την εντολή insertPatientRecord θεωρείται αυτονόητο ότι τα ορίσματα δίνονται με τη σωστή σειρά, χωρισμένα με κενά και στο τέλος της γραμμής υπάρχει\n.

Οι ημερομηνίες αποθηκεύονται και επεξεργάζονται σα string της μορφής DD-MM-YYYY. Έχει

υλοποιηθεί συνάρτηση που συγκρίνει δύο ημερομηνίες επιστρέφοντας int τιμές ανάλογα με το

ποια ημερομηνία είναι μεταγενέστερη, αν κάποια είναι "-" κλπ. Αν δοθεί λάθος ημερομηνία, η compareDates τερματίζει το πρόγραμμα.

Για διάφορες δομές όπως για τη λίστα, υλοποιήθηκαν κάποιες συναρτήσεις, οι οποίες εν τέλει δε

χρησιμοποιούνται και είναι σχολιασμένες.

Στα Assets υπάρχουν διάφορα αρχεία εισόδου.

AVL

Έστω ότι οι τιμές βάσει των οποίων εισάγονται και κατανέμονται νέοι κόμβοι στο AVL Δέντρο

ονομάζονται AVL κλειδιά.

Τα AVL κλειδιά που έχουν τιμή ίση με αυτή του τρέχοντος AVL κόμβου κατανέμονται προς τα

δεξιά παιδιά του, μαζί δηλαδή με τους AVL κόμβους που έχουν μεγαλύτερες τιμές AVL

κλειδιών από αυτόν.

Η συνάρτηση getUnhealed χρησιμοποιείται για να βρούμε αναδρομικά πόσοι ασθενείς είναι ακόμα

στο νοσοκομείο, δηλαδή έχουν στο exitDate "-".

Η συνάρτηση get_child_nodes χρησιμοποιείται για το querry diseaseFrequency και για την περίπτωση που δίνεται country και για την περίπτωση που δε δίνεται.

Το Balance Factor ενός κόμβου είναι η διαφορά του height του αριστερού παιδιού από το height του

δεξιού παιδιού.

Η συνάρτηση compareAdd κατεβαίνει το δέντρο για να βρει πού θα μπεί ο καινούριος κόμβος συγκρίνοντας τις ημερομηνίες των υπάρχουσων κόμβων με αυτόν. Αυτή καλεί και τον έλεγχο για πιθανά rotations.

Στην περιπτωση μόνο του δέντρου που ελέγχει για Duplicate κλειδιά, ο έλεγχος στους κόμβους γίνεται με βάση το recordId.

Αν εισήχθη καινούριος κόμβος στο AVL, γίνεται έλεγχος για rotations.

Η συνάρτηση UpdateExitDate ελέγχει το AVL Duplicate tree για να αλλάξει το exitDate σε ένα record αν

υπάρχει και αν είναι σωστό σε σύγκριση με το entryDate.

Η συνάρτηση για τα AVL performRotations ελέγχει αν μετά από μία εισαγωγή στο δέντρο απαιτείται

κάποιο rotation. Για ένα κόμβο existent ελέγχεται πρώτα για rotation το δεξί του υπόδεντρο και

έπειτα το αριστερό. Οι έλεγχοι που γίνονται με τις μεταβλητές balance, newcompare,

does_it_have_at_least_one_not_NULL_child εξασφαλίζουν ότι για να γίνει rotation στο

subtree ενός κόμβου πρέπει και το balance factor να έχει τιμή μικρότερη του -1 ή μεγαλύτερη του 1

και ο καινούριος κόμβος added να έχει εισαχθεί στο υπόδεντρο που ελέγχουμε αλλά και το child node του κόμβου existent να έχει τουλάχιστον ένα παιδί. Αν δεν έχει σημαίνει ότι το child node του existent

είναι αυτό που μόλις εισήχθη, οπότε δεν γίνεται κάποιο rotation στο συγκεκριμένο υπόδεντρο.

Είναι πιθανό τα rotations στην υλοποίησή μου να τα έχω ονομάσει αντίθετα σε σχέση με την πλειονότητα των παραδειγμάτων που υπάρχουν στο ίντερνετ, π.χ. LL αντί για RR κ.λ.π.

Η συνάρτηση recPrintAVLNode φτιάχτηκε μετά από μελέτη παρόμοιων συναρτήσεων εκτύπωσης

σε δενδρική μορφή.

Η ρίζα έχει nodeHeight το πλήθος των επιπέδων. Άρα το τελευταίο παιδί κάθε υποδέντρου έχει

nodeHeight 1. Το nodeHeight κάθε κόμβου του AVL χρησιμεύει μόνο για τα rotations. Δεν

είναι πάντα το σωστό γιατί μπορεί πχ το δέντρο κάποια στιγμή να είναι πλήρως ισορροπημένο και

μετά να εισαχθεί ένας κόμβος. Το nodeHeight του κόμβου και όλων των προκατόχων του θα αλλάξει

αλλά όχι και των γειτονικών υποδέντρων.

Hash Table:

Από Piazza: Στο bucket_size περιλαμβάνεται ο pointer στο επόμενο bucket. Πρέπει να δίνεται bucket_size τουλάχιστον 24 bytes για να χωράει 2 void pointers και ένας pointer σε επόμενο bucket.

Η hashFunction δέχεται ένα string και έναν ακέραιο διαιρέτη. Αθροίζει τις int τιμές κάθε χαρακτήρα

του int και επιστρέφει το υπόλοιπο της διαίρεσης του αθροίσματος με το διαιρέτη.

Στη συνάρτηση που δημιουργεί ένα Hash Table γίνεται έλεγχος για την παράμετρο bucketSize που

έχει δοθεί. Κάθε δομή Bucket περιέχει έναν πίνακα void pointer. Αυτοί οι pointers δουλεύουν κατά

ζεύγη. Ο ένας δείχνει στο όνομα του ιού/χώρας και ο επόμενος δείχνει στο AVL δέντρο που αποθηκεύει

τα αντίστοιχα records ασθενών. Το μέγεθος ενός void pointer είναι 8 bytes. Επίσης οι void pointers

πάνε αναγκαστικά κατά ζευγάρια, οπότε αν με δεδομένο ένα bucketSize υπάρχει υπόλοιπο στη διαίρεση bucketSize/16 bytes τότε αυτό ουσιαστικά δε χρησιμοποιείται.

Max Heap:

Για την υλοποίηση του Max Heap το id χρησιμοποιείται για να τοποθετηθούν οι κόμβοι στο δέντρο

διατηρώντας το χαρακτηριστικό του complete binary tree όπου οι κόμβοι μπαίνουν όσο πιο αριστερά

γίνεται σε κάθε επίπεδο. Το δέντρο κατανέμεται με βάση το πεδίο total κάθε κόμβου, που κρατάει

τον αριθμό φορών που βρέθηκε ένα country ή ένα disease. Όταν υπάρχει ανάγκη για reheapification,

οι δύο κάθε φορά εμπλεκόμενοι κόμβοι απλά ανταλλάσουν μεταξύ τους τα περιεχόμενα occurence και total.

Συνεπώς τα id μένουν πάντα στη θέση που πρωτοεισήχθησαν και έτσι είναι και στη συνέχεια η εισαγωγή νέων κόμβων.

Η συνάρτηση searchAllexisting αυξάνει το total των εμφανίσεων ενός κόμβου του Heap.

Η συνάρτηση compareFatherWithChild συγκρίνει έναν πατέρα με το παιδί του και αν χρειάζεται

ανταλλάζει τα περιεχόμενά τους.

Αν στα Querries, για το insertPatientRecord δοθεί record με Id που υπάρχει ήδη, εμφανίζεται αντίστοιχο μήνυμα, χωρίς όμως να τερματίζεται το πρόγραμμα όπως στην περίπτωση της αρχικής εισαγωγής records.