ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΡΧΕΙΩΝ

ΕΡΓΑΣΤΗΡΙΑΛΗ ΑΣΚΗΣΗ 2

ΔΥΑΔΥΙΚΑ ΔΕΝΤΡΑ ΑΝΑΖΗΤΗΣΗΣ ΚΑΙ ΟΥΡΕΣ ΠΡΩΤΕΡΑΙΟΤΗΤΑΣ

ΜΙΧΑΗΛ ΜΑΡΚΕΤΑΚΗΣ 2017030165

ΜΕΡΟΣ Α)

Binary Search Tree (Array implementation)

Για την υλοποίηση του δυαδικού δέντρου με Array(Στατικη δεσμευση μνήμης), χρησιμοποιήθηκαν 3 Arrays μεγεθους 10^6 το κάθε ένα,το πρώτο για την πληροφορία (key) του κάθε κόμβου του δέντρου,το δεύτερο για την τιμή του αριστερού υποδένδρου και το τρίτο για την τιμή του δεξιού. Αρχικά χρειάστηκε η αρχικοποίηση της στοίβας ή αλλιώς του πίνακα Right και αυτό έγινε στον constructor.

Η συνάρτηση insert αρχικά αρχικοποιούσε τη πρώτη γραμμή του πίνακα(info[0],left[0],right[0]) με τα στοιχεία info, left και right του πρώτου κλειδιού που εισάχθηκε το οποίο είναι το κλειδί του root το οποίο σε αυτήν τη περίπτωση ειναι πάντα το 0.

Μετά η insert καλεί την αναδρομική συνάρτηση setTheAdresses η οποία με τη βοήθεια της στοίβας και της τιμής AVAIL κάνει τη σωστή διάσχιση κάθε φορα καθώς και ενημερώνει το left/right του στοιχείου που είχε παιδί το νέο κλειδί.

Η συνάρτηση search κάνει διάσχιση του δέντρου ανάλογα με τη τιμή του κλειδιού και επιστρέφει την θέση όπου βρέθηκε το κλειδί(θέση του πίνακα Info) οταν το βρεί, και -1 σε περίπτωση όπου το κλειδί δεν υπάρχει στο δεντρο.

Η συνάρτηση DeleteKey παίρνει σαν όρισμα ένα κλειδί προς διαγραφή και το διαγράφει κάνοντας όλες τις απαραίτητες αλλαγές στο δέντρο.

Στην πιο απλή περίπτωση που ο κόμβος προς διαγραφή έχει μόνο ένα παιδί τότε γίνονται οι απαραίτητες αλλαγές,αλλα και στην περίπτωση όπου ο κόμβος έχει 2 παιδιά, καλείται η βπηθητική συνάρτηση minValue,ώστε να βρεθεί το αριστερότερο παιδί του δεξιού υποδένδρου που θα πάρει την θέση του κόμβου που θα διαγραφεί.

Η συνάρτησεις Insert,Search,DeleteKey λειτουργούν σε μεγάλο βαθμό αποτελεσματικά αφού έγιναν πολλά tests.

Binary Search Tree(Dynamic Implementation)

Υλοποιήθηκε μέσα από το web και πιο συγκεκριμένα:

https://www.geeksforgeeks.org/binary-search-tree-set-1-search-and-insertion/?ref=leftbar-rightbar

https://www.geeksforgeeks.org/binary-search-tree-set-2-delete/?ref=lbp

Μετρήσεις

Μεθοδος	Μέσος αριθμός Συγκρίσεων/εισαγωγη(10^6)	Συνολικός χρόνος για 10^6 εισαγωγές	Μέσος αριθμος συγκρίσεων/ανά διαγραφή (100 διαγραφές)	Συνολικός χρόνος για 100 διαγραφές
ΔΔΕ δυναμικα	26	5.3s	40	0.0012s
ΔΔΕ με array	25	5.3s	40	0.001s

Παρατηρούμε ότι οι διαφορές τόσο χρονικά όσο και στον μέσο αριθμό συγκρίσεων ανά εισαγωγή,διαγραφή για τις δύο περιπτώσεις μας είναι αρκετά κοντά.Αποδείξαμε ότι η στατική δέσμευση μνήμης συμφέρει πεισσότερο σε σχέση με την δυναμική,αφου οι συγκρίσεις είναι λιγότερες.Προφανώς,με ακόμα καλύτερη υλοποίηση του Array BST,Η η διαφορά στις μετρήσεις θα ήταν πολύ πιο ξεκάθαρη.

Μερος Β)

ΟΥΡΑ ΠΡΟΤΕΡΑΙΟΤΗΤΑΣ (Με χρήση Πινακα)

Στο δεύτερο μέρος της άσκησης υλοποίηθηκε η δομή δεδομένων ουρά προτεραιότητας heap . Ειδικότερα υλοποιήθηκε το Max Heap μιας και ένα εκ των ζητουμένων ήταν η διαγραφή του μέγιστου κλειδιού της ουράς. Η υλοποίηση με Array βρέθηκε από το ίντερνετ .Περιληπτικά στην υλοποίηση με array χρησιμοποιείται ένας ακέραιος πίνακας μεγέθους 10^6(το πλήθος των στοιχείων που επρόκειτω να εισαχθούν) .

Με την χρήση της συνάρτησης insert(int element) γίνεται η εισαγωγή ενός στοιχείου στην ουρά κάθε φορά,με την βοήθεια των βοηθητικής συνάρτησης swap(int fpos,int spos).

Η διαγραφή του μέγιστου κλειδιού από την ουρά γίνεται με χρήση της συνάρτησης

ExtractMax() η οποία καλεί την βοηθητική συνάρτιση maxHeapify(int pos) (για pos=0).

Η εισαγωγή των στοιχείων στην ουρά όταν αυτά δίνονται όλα μαζί γίνεται μέσω της συνάρτησης insertForQueueAll(int[] p) ,όπου όλα τα κλειδιά που δίνονται από την εκφώνηση τα βάζουμε σε έναν πίνακα ακεραίων μεγέθους 10^6 και έπειτα αυτόν τον πίνακα σαν όρισμα στην συνάρτηση μας.

Μέθοδος	Συνολικός χρόνος όταν τα κλειδιά δίνονται όλα μαζί	Συνολικός χρόνος όταν τα κλειδιά δίνονται ένα ένα	Μέσος αριθμός συγκρίσεων/ εισαγωγή	Συνολικός αριθμός συγκρίσεων/ διαγραφή	Συνολικός χρόνος για 100 διαγραφές
Ουρά προτεραιότητας με array	5s	0.007s	1	37	0.002s

*Όσο αφορά την ουρα προτεραιότητας με δυναμική παραχώρηση μνήμης,υλοποιήθηκαν μόνο η συνάρτηση και οι βοηθητικές συναρτήσεις για την εισαγωγή των κλειδιών στην ουρά(‘ενα στοιχείο ανά κλήση),η οποία μετά από test που έγιναν λειτουργεί αλλα για το αρχείο των 10^6 κλειδιών κάνει πολύ ώρα.