

Ονοματεπώνυμο: Αλέξανδρος Βεντούρας

(δεν συμμετείχα σε ομάδα στην εργασία 3)

A.M: 3160013

Η εργασία 3:

Μέθοδος διάσχισης:

Υλοποίησα την ενδοδιατεταγμένη διάσχιση, στην οποία αναδρομικά σε ένα δέντρο, επεξεργαζόμαστε πρώτα το αριστερό υποδέντρο, μετά τον κόμβο, και μετά το δεξί υποδέντρο (μέθοδοι `printTree` και `searchBook`).

Μέθοδοι εισαγωγής:

Μέθοδοι `insertWarehouse`, `insertR` και `insertT`:

Η μέθοδος `insertWarehouse` αυτή εισάγει ένα κόμβο – αποθήκη στο δέντρο – δίκτυο αποθηκών. Συγκεκριμένα, με την κλήση της `insertR` εισάγεται ο νέος κόμβος είτε στην ρίζα του δέντρου, είτε στο σημείο όπου θα τοποθετούνταν μέσα από την αναδρομική κλήση της `insertT`, βρίσκοντας το μονοπάτι όπου βρίσκεται το σημείο όπου πρέπει να εισαχθεί, ελέγχοντας κάθε φορά αν ο το κλειδί του κόμβου που θέλουμε να εισάγουμε είναι μεγαλύτερο ή μικρότερο από έναν κόμβο. Αν είναι μικρότερο, τότε επιλέγουμε το υποδέντρο στα αριστερά ενώ στην αντίθετη περίπτωση επιλέγουμε το υποδέντρο στα δεξιά. Ο αλγόριθμος τερματίζεται αν το υποδέντρο που επιλέχθηκε δεν υπάρχει (δηλαδή ο κόμβος δεν έχει το αντίστοιχο παιδί). Οπότε, ο νέος κόμβος εισάγεται σε αυτή τη θέση.

Η απόφαση για το αν θα γίνει η κλασσική εισαγωγή που περιγράφεται παραπάνω, ή αν θα γίνει εισαγωγή στην ρίζα του δέντρου, λαμβάνεται μέσα από μια συνθήκη που είναι αληθής ή ψευδής ανάλογα με το αποτέλεσμα της συνάρτησης παραγωγής τυχαίων `random()`.

Μέθοδος `insertBookAtWarehouse`:

Η μέθοδος `insertBookAtWarehouse` εισάγει ένα συγκεκριμένο αριθμό αντιγράφων ενός βιβλίου σε μία συγκεκριμένη αποθήκη.

Συγκεκριμένα, αναζητείται ο κόμβος-αποθήκη με το δωθέν `id`, και αν αυτός υπάρχει τότε: Αν υπάρχει ήδη το βιβλίο, προστίθεται στη σωστή θέση της λίστας βιβλίων της αποθήκης ο αριθμός των νέων αντιγράφων, αλλιώς δημιουργείται ένας νέος κόμβος στη λίστα βιβλίων της αποθήκης, και αρχικοποιείται με τα στοιχεία του βιβλίου.

Εφαρμόζεται η μέθοδος της αναζήτησης ενός κόμβου σε ΔΔΑ που περιγράφεται παρακάτω.

Μέθοδοι αναζήτησης:

Μέθοδοι `search` και `searchR`:

Η `search` καλεί τη `searchR`, η οποία βρίσκει το μονοπάτι όπου βρίσκεται η αποθήκη που αναζητούμε, ελέγχοντας κάθε φορά αν ο το κλειδί της αποθήκης είναι μεγαλύτερο, ίσο ή

μικρότερο από έναν κόμβο. Αν είναι μικρότερο, τότε επιλέγουμε το υποδέντρο στα αριστερά ενώ αν είναι μεγαλύτερο επιλέγουμε το υποδέντρο στα δεξιά και συνεχίζουμε αναδρομικά. Ο αλγόριθμος τερματίζεται αν το κλειδί της αποθήκης είναι ίσο με το κλειδί του κόμβου, ή αν το υποδέντρο που επιλέχθηκε δεν υπάρχει (δηλαδή ο κόμβος δεν έχει το αντίστοιχο παιδί), οπότε ο αναζητούμενος κόμβος δεν υπάρχει .

Μέθοδοι διαγραφής:

Η μέθοδος `removeBook`:

Η μέθοδος αυτή αφαιρεί ένα αντίγραφο ενός βιβλίου σε μία συγκεκριμένη αποθήκη.

Συγκεκριμένα, αναζητείται ο κόμβος-αποθήκη με το δωθέν id, και αν αυτός υπάρχει τότε: Αν υπάρχει ήδη το βιβλίο, μειώνεται κατά ένα ο αριθμός των νέων αντιγράφων στη σωστή θέση της λίστας των βιβλίων της αποθήκης, και αν έχουν εξαντληθεί όλα τα αντίτυπα, αφαιρείται το συγκεκριμένο βιβλίο από την λίστα .

Εφαρμόζεται η μέθοδος της αναζήτησης ενός κόμβου σε ΔΔΑ που περιγράφεται παραπάνω.

Οι μέθοδοι `removeWarehouse` και `removeR`:

Η μέθοδος `removeWarehouse` αυτή διαγράφει ένα κόμβο – αποθήκη από το δέντρο – δίκτυο αποθηκών. Συγκεκριμένα, με την κλήση της `removeR` αναζητείται και ανευρίσκεται ο υπό διαγραφή κόμβος, και έπειτα γίνονται οι κατάλληλες μετατροπές στη δομή του δέντρου, προκειμένου να μην διασπαστεί το δέντρο (περιστροφές κλπ).