

Η/Υ 432: Δομές Δεδομένων

Χειμερινό Εξάμηνο Ακαδημαϊκού Έτους 2006-2007

Παναγιώτα Φατούρου

4^ο Σετ Ασκήσεων

Ημερομηνία Παράδοσης: Πέμπτη, 21/12, ώρα 13:00-15:00.

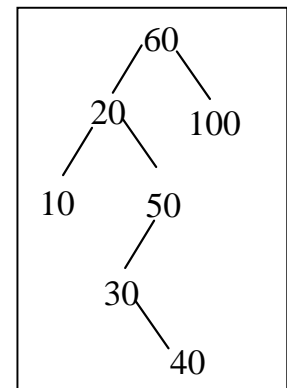
Τρόπος Παράδοσης: Οι ασκήσεις παραδίδονται στους βοηθούς του μαθήματος (Νικόλαος Καλλιμάνης, Ελευθέριος Κοσμάς και Μαρία Χριστοδουλίδου) το αργότερο μέχρι την ημερομηνία παράδοσης. Το γραφείο των βοηθών του μαθήματος είναι το Α25.

Άσκηση 1

- α. Εισάγετε τα στοιχεία με κλειδιά 10, 5 και 7 σε μια αρχικά άδεια λίστα παράλειψης (skip list). Υποθέστε ότι $\text{MaxLevel} = 4$ και θεωρήστε ότι η ακολουθία από bits 10111011110 καθορίζει το αν ο έλεγχος « $\text{Random}() < 1/2$ » στην $\text{RandomLevel}()$ επιστρέφει TRUE ή FALSE. Το i -οστό bit της παραπάνω ακολουθίας καθορίζει αν τη i -οστή φορά που θα εκτελεστεί ο έλεγχος θα αποτιμηθεί σε TRUE ή FALSE (αν το i -οστό bit είναι 1 ο έλεγχος θα αποτιμηθεί σε TRUE, διαφορετικά σε FALSE).
- β. Παρουσιάστε ψευδοκώδικα για τη λειτουργία $\text{MakeEmptySkipList}()$ που δημιουργεί μια κενή λίστα παράλειψης (skip list) και τη λειτουργία $\text{DeleteItemFromSkipList}(\text{int } x)$ που διαγράφει ένα στοιχείο από μια λίστα παράλειψης.

Άσκηση 2

- α. Ας υποθέσουμε ότι ένα ταξινομημένο δένδρο αποτελείται από τα κλειδιά 1 ως 100. Έστω ότι αναζητείται το κλειδί 50. Ποια/ες από τις ακόλουθες ακολουθίες δεν θα μπορούσε να είναι έγκυρη ακολουθία κόμβων που προσπελαύνονται κατά την αναζήτηση;
- 10, 30, 80, 90, 40, 60.
 - 10, 30, 80, 20, 60.
 - 10, 30, 80, 40, 60.
- β. Παρουσιάστε τα δένδρα που προκύπτουν κατά την εισαγωγή των κλειδιών 45, 15 και 12 στο δένδρο του σχήματος.
- β. Παρουσιάστε τα δένδρα που προκύπτουν κατά τη διαγραφή των κλειδιών 50, 20, 60 και 100 από το δένδρο που προκύπτει μετά τις εισαγωγές του ερωτήματος β.



Υπόδειξη: Στα ερωτήματα γ. και δ. θα πρέπει να παρουσιάσετε όλα τα δένδρα που προκύπτουν μετά από κάθε μία εισαγωγή ή διαγραφή.

Άσκηση 3

- α. Υποθέστε ότι δύο λεξικά υλοποιούνται ως ταξινομημένα δυαδικά δένδρα. Παρουσιάστε αλγόριθμο ο οποίος θα υλοποιεί τη λειτουργία $\text{BinarySearchTreeIntersection}(\text{struct treenode } *T1, \text{ struct treenode } *T2, \text{ struct treenode } *T3)$ όπου $T1$ είναι δείκτης στη ρίζα ενός δένδρου,

T2 είναι δείκτης στη ρίζα ενός άλλου δένδρου και T3 είναι δείκτης στη ρίζα ενός νέου δένδρου που θα δημιουργηθεί από την BinarySearchTreeIntersection() και **θα πρέπει να περιέχει τα κλειδιά που ανήκουν στην τομή των λεξικών T1 και T2. Ο αλγόριθμός σας θα πρέπει να έχει πολυπλοκότητα $O(|T1|+|T2|+|T3|*height(T3))$** (όπου |T1|, |T2| και |T3| είναι ο αριθμός στοιχείων των λεξικών T1, T2 και T3 αντίστοιχα, ενώ height(T3) είναι το ύψος του δένδρου T3 μετά την εισαγωγή όλων των κλειδιών της τομής των T1 και T2 σε αυτό). Για να επιτύχετε αυτήν την πολυπλοκότητα επιτρέπεται να χρησιμοποιήσετε λίστες (στοίβες, ουρές ή γενικές λίστες) ως βοηθητικές δομές. Επιτρέπεται επίσης να δημιουργήσετε όσες βοηθητικές συναρτήσεις χρειάζεστε.

- β. Περιγράψτε αλγόριθμο, ο οποίος θα παίρνει ως όρισμα έναν δείκτη σε έναν κόμβο ενός (ταξινομημένου) διπλά-συνδεδεμένου δυαδικού δένδρου και θα επιστρέφει τον επόμενο του κόμβου στην μετα-διατεταγμένη διάσχιση.