

Η/Υ 432: Δομές Δεδομένων

Χειμερινό Εξάμηνο Ακαδημαϊκού Έτους 2006-2007

Παναγιώτα Φατούρου

3^ο Σετ Ασκήσεων

Ημερομηνία Παράδοσης: Τρίτη, 12/12, ώρα 15:00-16:00.

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. Γράψτε μια συνάρτηση `RemoveRightChild(struct treenode *r)` η οποία θα παίρνει σαν όρισμα έναν δείκτη `r` στη ρίζα ενός δυαδικού δένδρου που αναπαριστά (προσομοιώνει) ένα (όχι απαραίτητα δυαδικό) διατεταγμένο δένδρο. Θεωρήστε ότι κάθε κόμβος του δένδρου αποθηκεύει εκτός από τους απαραίτητους δείκτες (`lc` και `rs`) και έναν ακέραιο `num`. Η συνάρτηση θα πρέπει να διασχίζει όλους τους κόμβους του διατεταγμένου δένδρου και για κάθε κόμβο θα πρέπει να διπλασιάζει την τιμή του δεξιότερου παιδιού του κόμβου.
- β. Γράψτε μια συνάρτηση `ChangeLeftWithRightChild(struct treenode *r)`, η οποία θα παίρνει σαν παράμετρο έναν δείκτη στη ρίζα ενός δυαδικού δένδρου και θα κάνει τα εξής: διασχίζει κάθε κόμβο του δένδρου και αλλάζει το δείκτη `lc` του κόμβου να δείχνει στο δεξί παιδί του κόμβου, καθώς και το δείκτη `rc` να δείχνει στο αριστερό παιδί του κόμβου. Στο τέλος της εκτέλεσης του αλγορίθμου σας, θα πρέπει το αριστερό παιδί κάθε κόμβου να έχει γίνει δεξιό και αντίστροφα.

Υπόδειξη: Είναι χρήσιμο (παρότι δεν ζητείται) να κάνετε `trace` την εκτέλεση των αλγορίθμων που θα σχεδιάσετε.

Άσκηση 2

Δίνεται ένα δυαδικό δένδρο με 14 κόμβους και κλειδιά A, B, C, D, E, F, G, H, I, J, K, L, M, N. Είναι γνωστό πως η ενδοδιατεταγμένη διάσχιση των κόμβων του δένδρου δίνει τα κλειδιά τους με την εξής διάταξη: C, E, I, A, D, H, L, B, G, M, J, K, N, F, ενώ η μεταδιατεταγμένη διάσχιση δίνει την διάταξη C, A, H, D, I, L, E, M, K, N, J, F, G, B στα κλειδιά των κόμβων. Ζωγραφίστε το δένδρο. Για κάθε κόμβο εξηγήστε βάσει ποιου σκεπτικού αποφασίσατε να τον ζωγραφίσετε στη θέση που βρίσκεται στο δένδρο που σχεδιάσατε.

Άσκηση 3

Σας ζητείται να γράψετε μια ρουτίνα (σε ψευδοκώδικα) η οποία θα παίρνει ως όρισμα έναν δείκτη στη ρίζα ενός δυαδικού δένδρου και θα τυπώνει τους κόμβους του δένδρου ανά επίπεδα ξεκινώντας ωστόσο από το βαθύτερο (χαμηλότερο) επίπεδο του δένδρου και όχι από το επίπεδο 0. Έτσι αν ένα δένδρο έχει βάθος k , οι κόμβοι του επιπέδου k θα πρέπει να τυπωθούν πρώτοι, στη συνέχεια οι κόμβοι του επιπέδου $k-1$, κ.ο.κ. Ο αλγόριθμος τερματίζει όταν τυπωθεί και ο κόμβος ρίζα που είναι ο μοναδικός κόμβος του επιπέδου 0. Η σειρά με την οποία τυπώνονται οι κόμβοι σε κάθε επίπεδο μπορεί να είναι αυθαίρετη (δηλαδή επιτρέπεται π.χ., οι κόμβοι να τυπώνονται

από δεξιά προς αριστερά). Στο παρακάτω σχήμα παρουσιάζεται ένα παράδειγμα δυαδικού δένδρου.

Η έξοδος του αλγορίθμου όταν αυτός εκτελείται με παράμετρο έναν δείκτη στη ρίζα του δένδρου του σχήματος θα πρέπει να είναι:

Κόμβοι Επιπέδου 4: 7

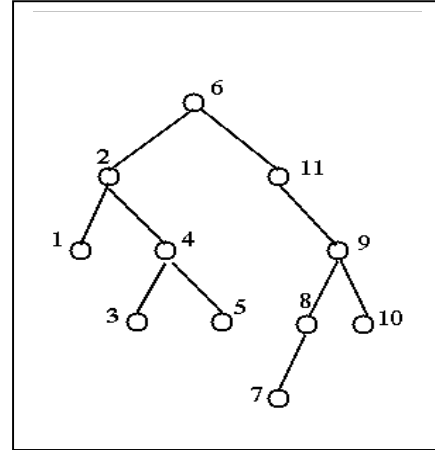
Κόμβοι Επιπέδου 3: 10, 8, 5, 3

Κόμβοι Επιπέδου 2: 9, 4, 1

Κόμβοι Επιπέδου 1: 11, 2

Κόμβοι Επιπέδου 0: 6

Το k δεν είναι γνωστό (δηλαδή θα πρέπει ο αλγόριθμός σας να το υπολογίζει). Προσέξτε ότι ο αλγόριθμος που ζητείται είναι πιο πολύπλοκος από διάσχιση ανά επίπεδα για τους εξής λόγους: (1) είναι ένα είδος αντίστροφης διάσχισης ανά επίπεδα (αφού οι κόμβοι των χαμηλότερων επιπέδων τυπώνονται πρώτοι), και (2) θα πρέπει ο αλγόριθμος να γνωρίζει και την επιπρόσθετη πληροφορία του ποιοι κόμβοι ανήκουν σε ποιο επίπεδο.



Υπόδειξη: Χρησιμοποιήστε μια ουρά και μια στοίβα. Χρησιμοποιήστε επίσης μια ειδική τιμή (π.χ. το null), την οποία θα εισάγεται στην ουρά (ίσως και στη στοίβα) κάθε φορά που όλοι οι κόμβοι ενός επιπέδου έχουν εισαχθεί σε αυτή. Έτσι, θα κρατάτε χρήσιμη πληροφορία για το ποιοι κόμβοι ανήκουν στα διάφορα επίπεδα.