

HY240: Δομές Δεδομένων **Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2008-09** **Παναγιώτα Φατούρου**

3^ο Σετ Ασκήσεων

Ημερομηνία Παράδοσης: Παρασκευή, 14 Νοεμβρίου 2008

Τρόπος Παράδοσης: Στο εργαστήριο των μεταπτυχιακών στο υπόγειο των λευκών κτιρίων, ώρα 12:00-13:00. Ασκήσεις που παραδίδονται μετά τις 13:00 θεωρούνται εκπρόθεσμες. Οι φοιτητές μπορούν να παραδίδουν τις ασκήσεις τους και νωρίτερα από αυτή την ημερομηνία αλλά δικαιολογίες όπως «ήρθα την τάδε μέρα και δεν ήσασταν στο γραφείο σας» δεν θα γίνουν δεκτές. Όσοι από τους φοιτητές δεν έχουν καταφέρει να παραδώσουν τις ασκήσεις τους πριν τις 14/11 θα πρέπει να τις παραδώσουν στο εργαστήριο των μεταπτυχιακών την προκαθορισμένη μέρα και ώρα. Οι φοιτητές μπορούν να παραδίδουν τις ασκήσεις τους ηλεκτρονικά αλλά σε αυτή την περίπτωση απαιτείται να γίνουν scan οι λύσεις και είτε να χρησιμοποιηθεί το πρόγραμμα submit (σύμφωνα με τις οδηγίες που στάλθηκαν από τους βοηθούς του μαθήματος) ή να σταλούν μέσω ηλεκτρονικού ταχυδρομείου στο λογαριασμό hy240@csd.uoc.gr. Αυτός είναι και ο τρόπος παράδοσης εκπρόθεσμων ασκήσεων.

Οι υποχρεωτικές ασκήσεις αυτής της σειράς είναι οι 1-4 και βαθμολογούνται με άριστα το 100. Φοιτητές που θα επιλύσουν σωστά επιπρόσθετα των ασκήσεων 1-4 και την άσκηση 5 θα βαθμολογηθούν με το βαθμό 125/100 (η τεχνική των ασκήσεων bonus δίνει τη δυνατότητα στους φοιτητές να βελτιώσουν τη συνολική τους βαθμολογία στις σειρές θεωρητικές ασκήσεων).

1. α. Παρουσιάστε τα δένδρα που προκύπτουν κατά την εισαγωγή των ακόλουθων κλειδιών (με τη σειρά που δίνονται) σε ένα αρχικά άδειο ταξινομημένο δυαδικό δένδρο: 50, 25, 75, 15, 17, 6, 5, 7, 30, 41, 36, 38, 37.

- β. Παρουσιάστε τα δένδρα που προκύπτουν κατά τη διαγραφή των κλειδιών 25, 75, 17 και 38 από το δένδρο που προκύπτει μετά τις εισαγωγές του ερωτήματος α.

Σημείωση: Θα πρέπει να παρουσιάσετε όλα τα δένδρα που προκύπτουν μετά από κάθε μία εισαγωγή ή διαγραφή.

2. Είναι γνωστό ότι η ταξινόμηση n στοιχείων απαιτεί χρόνο $\Omega(n \log n)$ όταν ο αλγόριθμος ταξινόμησης εκτελείται βασιζόμενος μόνο σε συγκρίσεις (δηλαδή γνωρίζουμε ότι δεν υπάρχει αλγόριθμος ταξινόμησης, βασιζόμενος σε συγκρίσεις, που να έχει πολυπλοκότητα μικρότερη από τάξης $n \log n$). Επιχειρηματολογήστε πως κάθε αλγόριθμος για την κατασκευή ενός ταξινομημένου δυαδικού δένδρου αναζήτησης από μια αυθαίρετη λίστα n στοιχείων απαιτεί χρόνο $\Omega(n \log n)$.

3. α. Γράψτε μια διαδικασία `RemoveRightChild( pointer 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 διαγράφει το δεξιότερο παιδί του κόμβου στο διατεταγμένο δένδρο.

Υπόδειξη: Η διαγραφή ενός κόμβου v έχει ως συνεπακόλουθο τη διαγραφή όλου του υπο-δένδρου που εκφύεται από τον v .

- β. Γράψτε μια διαδικασία `ChangeLeftWithRightChild( pointer 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.
- γ. Γράψτε μια συνάρτηση `IsPerfectTree(pointer r)`, η οποία θα παίρνει σαν παράμετρο έναν δείκτη στη ρίζα ενός δυαδικού δένδρου και θα επιστρέφει `TRUE` αν το δένδρο είναι τέλειο και `FALSE` διαφορετικά.

Υπόδειξη: Είναι χρήσιμο (παρότι δεν ζητείται) να κάνετε trace την εκτέλεση των αλγορίθμων που θα σχεδιάσετε. Κάθε μια από τις διαδικασίες/συναρτήσεις που ζητούνται επιτρέπεται να καλούν άλλες συναρτήσεις, αν θεωρείτε ότι αυτό είναι χρήσιμο.

4. Δίνεται ένα δυαδικό δένδρου 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.

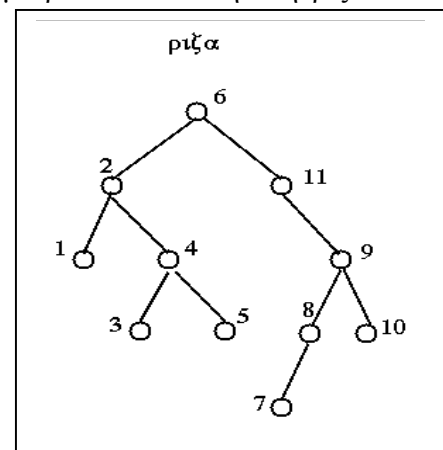
Άσκηση Bonus

Η άσκηση αυτή είναι προαιρετική και δίνει 25 επιπρόσθετες μονάδες για βελτίωση της συνολικής βαθμολογίας όλων των σειρών των θεωρητικών ασκήσεων.

5. 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 τον αλγόριθμο διάσχισης 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.