

HY240: Δομές Δεδομένων
Εαρινό Εξάμηνο – Ακαδημαϊκό Έτος 2016-17
Παναγιώτα Φατούρου

3^ο Σετ Ασκήσεων

Ημερομηνία Παράδοσης: Δευτέρα, 24 Απριλίου 2017

Τρόπος Παράδοσης: Οι ασκήσεις μπορούν να παραδοθούν είτε σε ηλεκτρονική μορφή χρησιμοποιώντας το πρόγραμμα turnin (δείτε οδηγίες στην ιστοσελίδα του μαθήματος), είτε στους βοηθούς του μαθήματος στο εργαστήριο των μεταπτυχιακών, τη Δευτέρα, 24 Απριλίου 2017, ώρα 13:00-14:00. Ασκήσεις που παραδίδονται μετά τις 14:00 της Δευτέρας, 24/4/2017 θεωρούνται εκπρόθεσμες. Εκπρόθεσμες ασκήσεις γίνονται δεκτές σε ηλεκτρονική μορφή και η αποστολή τους πρέπει να γίνει χρησιμοποιώντας το πρόγραμμα turnin. Εναλλακτικά, εκπρόθεσμες ασκήσεις μπορούν να παραδοθούν στη διδάσκουσα.

Άσκηση 1 [40 μονάδες]

α. Θεωρήστε τη λειτουργία $\text{Intersection}(S_1, S_2)$ (τομή), όπου S_1 και S_2 είναι δύο σύνολα. Παρουσιάστε αλγόριθμο που να υλοποιεί την $\text{Intersection}()$, δεδομένου ότι το S_1 υλοποιείται με ένα **ταξινομημένο δένδρο** (δηλαδή με ένα δένδρο δυαδικής αναζήτησης) και το S_2 **υλοποιείται με μια ταξινομημένη, απλά-συνδεδεμένη λίστα**. Η $\text{Intersection}()$ πρέπει να παίρνει ως όρισμα ένα δείκτη στη ρίζα του δένδρου που υλοποιεί το σύνολο S_1 κι ένα δείκτη στο πρώτο στοιχείο της λίστας που υλοποιεί το S_2 και θα πρέπει να τυπώνει όλα τα κλειδιά του S_1 που υπάρχουν και στο S_2 . Τόσο το δένδρο που αναπαριστά το S_1 , όσο και η λίστα που αναπαριστά το S_2 θα πρέπει να παραμένουν αναλλοίωτα κατά την εκτέλεση του αλγορίθμου.

Ο αλγόριθμός σας θα πρέπει να έχει πολυπλοκότητα $O(n_1 + n_2)$, όπου n_1 και n_2 είναι οι πληθικοί αριθμοί των συνόλων S_1 και S_2 , αντίστοιχα. Ο αλγόριθμός σας δεν επιτρέπεται να χρησιμοποιεί βοηθητικές δομές δεδομένων (π.χ. λίστες, πίνακες, κλπ.) δηλαδή η τομή των δύο συνόλων θα πρέπει να υπολογίζεται καθώς γίνεται η διάσχιση του δένδρου. Εξηγήστε γιατί ο αλγόριθμός σας επιτυγχάνει τη ζητούμενη πολυπλοκότητα αυτή. [20M]

β. Παρουσιάστε αλγόριθμο ο οποίος θα ελέγχει αν ένα δυαδικό δένδρο είναι AVL. [20M]

Υπόδειξη: Για να απαντήσετε τα παραπάνω ερωτήματα επιτρέπεται να δημιουργήσετε περισσότερες της μιας συναρτήσεις, τις οποίες μπορείτε να καλείτε από την πρωταρχική συνάρτηση που υλοποιεί τον αλγόριθμό σας.

Άσκηση 2 [30 Μονάδες]

α. Δίνεται ένα δένδρο T που είναι κοκκινόμαυρο. Παρουσιάστε αλγόριθμο ο οποίος θα παίρνει ως παράμετρο έναν δείκτη στη ρίζα του T και θα υπολογίζει **το μαύρο ύψος** του δένδρου. Ο αλγόριθμός σας θα πρέπει να έχει χρονική πολυπλοκότητα $O(\log n)$, όπου n είναι το πλήθος των κόμβων του δένδρου. Θεωρήστε ότι κάθε κόμβος είναι ένα struct με πεδία key, info, color, lc και rc. [15M]

β. Δίνεται ένα δένδρο T που είναι AVL. Παρουσιάστε αλγόριθμο ο οποίος θα παίρνει ως παράμετρο έναν δείκτη στη ρίζα του T και θα υπολογίζει **το ύψος** του δένδρου. Ο αλγόριθμός σας θα πρέπει να έχει χρονική πολυπλοκότητα $O(\log n)$, όπου n είναι το πλήθος των κόμβων του δένδρου. Θεωρήστε ότι κάθε κόμβος είναι ένα struct με πεδία key, info, balance, lc και rc. [15M]

Άσκηση 3 [30 μονάδες]

α. Περιγράψτε αλγόριθμο ο οποίος, δοθέντος ενός φύλλου v , θα βρίσκει τον κόμβο που περιέχει το κοντινότερο μεγαλύτερο κλειδί σε εκείνο του v μεταξύ των κλειδιών όλων των κόμβων του δένδρου.

[15M]

β. Ένα δυαδικό μετα-διατεταγμένο αντιταξινομημένο δέντρο είναι ένα δυαδικό δέντρο για το οποίο η μετα-διατεταγμένη διάσχιση δίνει τα κλειδιά που είναι αποθηκευμένα στους κόμβους του δένδρου σε φθίνουσα διάταξη. Παρουσιάστε ψευδοκώδικα ο οποίος θα βρίσκει αν υπάρχει κόμβος με κλειδί k σε δυαδικό μεταδιατεταγμένο αντιταξινομημένο δένδρο (δηλαδή ζητείται αλγόριθμος αναζήτησης σε τέτοια δένδρα). Ο αλγόριθμός σας θα πρέπει να έχει χρονική πολυπλοκότητα $O(h)$, όπου h είναι το ύψος του δένδρου.

[15M]