

HY240: Δομές Δεδομένων
Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2017-18
Παναγιώτα Φατούρου

3^ο Σετ Ασκήσεων

Ημερομηνία Παράδοσης: Δευτέρα, 4 Δεκεμβρίου 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, τη Δευτέρα, 4 Δεκεμβρίου 2017, ώρα 15:00-16:00. Ασκήσεις που παραδίδονται μετά τις 16:00 της Δευτέρας, 4/12/2017 θεωρούνται εκπρόθεσμες. Εκπρόθεσμες ασκήσεις γίνονται δεκτές σε ηλεκτρονική μορφή και η αποστολή τους πρέπει να γίνει χρησιμοποιώντας το πρόγραμμα turnin.

Άσκηση 1 [40 μονάδες]

- α. Θεωρήστε ένα απλά-συνδεδεμένο δυαδικό δένδρο T (χωρίς κόμβο φρουρό). Παρουσιάστε ψευδοκώδικα για μια συνάρτηση η οποία θα παίρνει ως όρισμα ένα δείκτη στη ρίζα του T και θα δημιουργεί ένα αντίγραφο του T . Η μόνη διαφορά του αντιγράφου του T από το ίδιο το T είναι πως το αντίγραφο θα πρέπει να είναι δένδρο με κόμβο φρουρό. Το αντίγραφο (καθώς και οι δείκτες προς τον κόμβο φρουρό) θα πρέπει να δημιουργούνται με μια μόνο διάσχιση του T . [20%]
- β. Θεωρήστε ένα δυαδικό δένδρο T που υλοποιεί ένα (όχι-απαραίτητα δυαδικό) διατεταγμένο δένδρο Δ για το οποίο ισχύουν τα εξής: το κλειδί κάθε κόμβου είναι διαφορετικό από τα κλειδιά όλων των υπόλοιπων κόμβων του δένδρου (δηλαδή τα κλειδιά των κόμβων είναι μοναδικά). Παρουσιάστε ψευδοκώδικα για αναδρομική συνάρτηση η οποία θα παίρνει, ως ορίσματα, έναν δείκτη στη ρίζα του T και έναν δείκτη σε κάποιον κόμβο v του T και θα επιστρέφει το βάθος του v στο διατεταγμένο δένδρο Δ (προσοχή, το βάθος αυτό δεν είναι απαραίτητα το ίδιο με το βάθος του v στο δυαδικό δένδρο T). [20%]

Σημείωση: Στη παράδειγμα της διαφάνειας 20 της Ενότητας 3, το βάθος που θα πρέπει ο αλγόριθμος να τυπώνει π.χ. για το κλειδί G είναι 2 που είναι το βάθος του στο δένδρο (A) και όχι 4 που είναι το βάθος του στο δένδρο (B). Η συνάρτηση επιτρέπεται να έχει και επιπρόσθετα ορίσματα.

Άσκηση 2 [30 μονάδες]

Θεωρήστε δύο ταξινομημένα δυαδικά δένδρα (δηλαδή δύο δένδρα δυαδικής αναζήτησης), έστω T_1 και T_2 . Ζητείται αλγόριθμος ο οποίος θα συγχωνεύει τα δύο δένδρα σε ένα σε χρόνο $O(n_1 + n_2)$, όπου n_1 και n_2 είναι το πλήθος των κόμβων του T_1 και του T_2 , αντίστοιχα. Το τελικό δένδρο θα πρέπει να έχει ύψος $O(\log(n_1+n_2))$.

Υπόδειξη: Τοποθετήστε τα στοιχεία του T_1 σε έναν πίνακα, ο οποίος θα είναι ταξινομημένος σε αύξουσα διάταξη, σε χρόνο $O(n_1)$. Τοποθετήστε τα στοιχεία του T_2 σε έναν δεύτερο πίνακα, ο οποίος θα είναι επίσης ταξινομημένος σε αύξουσα διάταξη, σε χρόνο $O(n_2)$. Συγχωνεύστε τα στοιχεία των δύο πινάκων σε έναν τρίτο πίνακα ο οποίος θα είναι επίσης ταξινομημένος σε αύξουσα διάταξη. Η συγχώνευση θα πρέπει να πραγματοποιείται σε χρόνο $O(n_1+n_2)$. Βάσει του τρίτου πίνακα, φτιάξτε το δένδρο T που θα περιέχει και τα στοιχεία του T_1 και εκείνα του T_2 . Το T θα πρέπει να έχει ύψος $O(\log(n_1+n_2))$ και η δημιουργία του θα πρέπει να πραγματοποιείται σε χρόνο $O(n_1+n_2)$.

Άσκηση 3 [40 Μονάδες]

- α. Θεωρήστε ένα ταξινομημένο, διπλά-συνδεδεμένο δυαδικό δένδρο T (δηλαδή το T είναι δένδρο δυαδικής αναζήτησης). Θεωρήστε επίσης μια απλά-συνδεδεμένη λίστα L η οποία είναι ταξινομημένη σε αύξουσα διάταξη. Έστω ότι **το σύνολο των κλειδιών της L είναι υποσύνολο του συνόλου των κλειδιών του T_1** . Παρουσιάστε ψευδοκώδικα για μια συνάρτηση η οποία για κάθε κόμβο v της L , θα διαγράφει από το T_1 τον κόμβο που έχει κλειδί ίδιο με εκείνο του v . Ο συνολικός χρόνος για να πραγματοποιηθούν όλες οι διαγραφές από το T_1 θα πρέπει να είναι $O(n)$, όπου n είναι το πλήθος των κόμβων του T_1 . [20%]

Σημείωση: Πρέπει να επιλέξετε την κατάλληλη διάσχιση για το T_1 ώστε να μπορέσετε να πραγματοποιήσετε σωστά όλες τις διαγραφές κόμβων με μία μόνο διάσχιση του T_1 (καθώς και μία μόνο διάσχιση της L).

- β. Θεωρήστε ένα ταξινομημένο δυαδικό δένδρο T (δηλαδή ένα δένδρο δυαδικής αναζήτησης). Έστω ότι κάθε κόμβος v του T περιέχει ένα πεδίο `cnt` το οποίο είναι ένας ακέραιος που εκφράζει τον αριθμό των κόμβων που υπάρχουν στο αριστερό υποδένδρο του v .

Παρουσιάστε ψευδοκώδικα για μια συνάρτηση η οποία θα παίρνει ως όρισμα έναν δείκτη στη ρίζα του T και έναν πίνακα B από bits και θα κάνει τα εξής. Θα εξετάζει τα bits με τη σειρά και, ξεκινώντας από τη ρίζα του T , κάθε φορά που το τρέχον bit είναι 0, θα ακολουθεί τον `lc` δείκτη του τρέχοντος κόμβου στο δένδρο, ενώ αν το bit είναι 1, θα ακολουθεί τον `rc` δείκτη του κόμβου. Ο αλγόριθμος θα σταματά όταν θα φτάνει στον τελευταίο κόμβο, έστω v , του μονοπατιού που ορίζεται από την ακολουθία bit B . Η συνάρτηση θα πρέπει να επιστρέφει πόσοι κόμβοι περιέχουν μικρότερα κλειδιά από εκείνο του κόμβου v . Ο αλγόριθμος θα πρέπει να εκτελείται σε $O(h)$ χρόνο, όπου h είναι το ύψος του T . [20%]