

HY240: Δομές Δεδομένων
Εαρινό Εξάμηνο – Ακαδημαϊκό Έτος 2018-19
Παναγιώτα Φατούρου

3^ο Σετ Ασκήσεων

Ημερομηνία Παράδοσης: Δευτέρα, 15 Απριλίου 2019

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, 15 Απριλίου 2019, ώρα 13:00-14:00. Ασκήσεις που παραδίδονται μετά τις 14:00 της Δευτέρας, 15/4/2019 θεωρούνται εκπρόθεσμες. Εκπρόθεσμες ασκήσεις γίνονται δεκτές σε ηλεκτρονική μορφή (pdf) και η αποστολή τους πρέπει να γίνει χρησιμοποιώντας το πρόγραμμα turnin. Εναλλακτικά, εκπρόθεσμες ασκήσεις μπορούν να παραδοθούν στη διδάσκουσα.

Άσκηση 1 [40 μονάδες]

- α. Δύο δυαδικά δένδρα λέγονται παρόμοια αν και τα δύο είναι κενά ή αν είναι παρόμοια τα αριστερά και τα δεξιά τους υποδένδρα, αντίστοιχα. Παρουσιάστε μια αναδρομική διαδικασία, η οποία, θα εξετάζει αν δύο δένδρα είναι παρόμοια. [15%]
- β. Παρουσιάστε αλγόριθμο που θα ελέγχει αν ένα δυαδικό δένδρο είναι AVL. [15%]
- γ. Παρουσιάστε αλγόριθμο ο οποίος δοθέντος ενός δείκτη σε κάποιο κόμβο v ενός ταξινομημένου δυαδικού δένδρου T , θα επιστρέφει το βάθος του v . Ο αλγόριθμός σας θα πρέπει να έχει χρονική πολυπλοκότητα $O(h)$, όπου h είναι το ύψος του δένδρου. [10%]

Άσκηση 2 [40 μονάδες]

Θεωρήστε την επέκταση του αφηρημένου τύπου δεδομένων Λεξικού που υποστηρίζει επιπρόσθετα τις εξής λειτουργίες: [Split: 20M, Merge: 20M]

- i. Split(D, k, D_1, D_2) η οποία δημιουργεί δύο νέα λεξικά, τα D_1 και D_2 , έτσι ώστε το D_1 να περιέχει τα κλειδιά του αρχικού λεξικού D που είναι μικρότερα ή ίσα το κλειδί k και το D_2 να περιέχει τα κλειδιά του D που είναι μεγαλύτερα του k .
- ii. Merge(D_1, D_2) η οποία συγχωνεύει τα λεξικά D_1 και D_2 σε ένα.

Παρουσιάστε ψευδοκώδικα που θα υλοποιεί τις παραπάνω λειτουργίες δεδομένου πως χρησιμοποιούνται δένδρα δυαδικής αναζήτησης (ταξινομημένα δένδρα) για την υλοποίηση των λεξικών (δηλαδή τα D_1, D_2, D και το λεξικό που προκύπτει από τη Merge είναι δένδρα δυαδικής αναζήτησης).

Η υλοποίηση της Split() θα πρέπει να έχει χρονική πολυπλοκότητα $O(h)$ όπου h είναι το ύψος του D . Η υλοποίηση της Merge() θα πρέπει να έχει πολυπλοκότητα $O(n_1+n_2)$ όπου n_1 και n_2 είναι το πλήθος των στοιχείων των D_1 και D_2 , αντίστοιχα.

Υπόδειξη: i) Για την Split(), εκτελέστε μια αναζήτηση για το k , αποκόβοντας τους δείκτες που βρίσκονται αριστερά και δεξιά του μονοπατιού. Τα υποδένδρα που προκύπτουν από την αποκοπή θα πρέπει να συνενώνονται κατάλληλα στα δένδρα που αναπαριστούν τα D_1 και D_2 , αντίστοιχα.

ii) Για την Merge(), πρέπει να δημιουργήσετε μια αναδρομική διαδικασία η οποία θα σαρώνει τα δένδρα (βάσει της ενδοδιατεταγμένης διάσχισης) και θα παράγει το ενιαίο δένδρο. Αν σας εξυπηρετεί, χρησιμοποιήστε βοηθητικές δομές για να αποθηκεύσετε τα κλειδιά των δένδρων, συγχωνεύστε τα κλειδιά σε βοηθητική δομή και στη συνέχεια φτιάξτε το συγχωνευμένο δένδρο.

Άσκηση 3 [20 Μονάδες]

Παρουσιάστε **μη αναδρομικό** αλγόριθμο που θα εκτελεί προ-διατεταγμένη διάσχιση σε ένα απλό δυαδικό δένδρο. Ο αλγόριθμός σας επιτρέπεται να χρησιμοποιεί μια επιπρόσθετη δομή στοίβας για να πραγματοποιήσει τη διάσχιση. [20%]