

HY240: Δομές Δεδομένων
Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2014-15
Παναγιώτα Φατούρου

4^ο Σετ Ασκήσεων

Ημερομηνία Παράδοσης: Τετάρτη, 7 Ιανουαρίου 2015

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, την Τετάρτη, 7 Ιανουαρίου 2015, ώρα 15:00-16:00. Ασκήσεις που παραδίδονται μετά τις 16:00 της Τετάρτης, 7/1/2015, 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 του μαθήματος ή να σταλούν στο hy240a@csl.uoc.gr μέσω ηλεκτρονικού ταχυδρομείου.

Άσκηση 1 [35 μονάδες]

α. Θεωρήστε ένα AVL δένδρο T που σε κάθε κόμβο του u είναι αποθηκευμένος ο αριθμός των κόμβων του αριστερού υποδένδρου του u . Παρουσιάστε αλγόριθμο (σε μορφή ψευδοκώδικα) ο οποίος θα υλοποιεί την παρακάτω λειτουργία σε χρόνο $O(\log n)$, όπου n είναι το πλήθος των κόμβων του δένδρου:

- `Access(T,i)`: επιστρέφει το i -οστό μικρότερο στοιχείο του δένδρου. [15%]

β. Θεωρήστε μια υλοποίηση ξένων συνόλων με ένωση, όπου μπορεί να υπάρχουν το πολύ `MAX_SETS` ξένα σύνολα. Η υλοποίηση χρησιμοποιεί έναν πίνακα κατακερματισμού `HT[0..MAX_POS-1]` στον οποίο αποθηκεύονται κλειδιά βάσει της μεθόδου ταξινομημένου κατακερματισμού με ανοικτή διεύθυνση (`ordered double hashing`). Έστω ότι h_1 και h_2 είναι η πρωτεύουσα και η δευτερεύουσα συνάρτηση κατακερματισμού, αντίστοιχα. Ο `HT` περιέχει τα κλειδιά των κόμβων όλων των πάνω δένδρων, καθώς και ένα δείκτη στον αντίστοιχο κόμβο για κάθε ένα από αυτά. Παρουσιάστε ψευδοκώδικα για τη συνάρτηση `UpTreeUnion` η οποία παίρνει ως όρισμα τα κλειδιά δύο κόμβων και συνενώνει τα πάνω δένδρα στα οποία ανήκουν οι κόμβοι (οι κόμβοι μπορεί να ανήκουν σε οποιαδήποτε πάνω δένδρα, ακόμη και στο ίδιο, στην οποία περίπτωση θα πρέπει να εμφανίζεται κατάλληλο μήνυμα). Κατά τη συνένωση, θα πρέπει να εφαρμόζονται οι τεχνικές συμπίεσης μονοπατιού και μείωσης ύψους. [20%]

Σημείωση: Θεωρήστε δεδομένες τις συναρτήσεις h_1 και h_2 .

Άσκηση 2 [45 μονάδες]

α. Παρουσιάστε αλγόριθμο (υπό τη μορφή ψευδοκώδικα) που θα διαγράφει ένα οποιοδήποτε στοιχείο ενός σωρού H σε χρόνο $O(\log n)$ (δηλαδή ζητείται μια υλοποίηση της `HearDelete(H, j)` η οποία θα διαγράφει το στοιχείο με δείκτη j από τον σωρό H σε χρόνο $O(\log n)$). [25%]

β. Σχεδιάστε εκ νέου τη λειτουργία `HearInsert()` ενός σωρού ώστε αυτή να διενεργείται από πάνω προς τα κάτω. Συγκεκριμένα, η `HearInsert()` θα πρέπει να κάνει κατάλληλα `swap` κλειδιών καθώς διασχίζει κόμβους από τη ρίζα προς τον κόμβο που θα αποτελέσει τον

πατρικό κόμβο του νεοεισαχθέντα κόμβου. Οι ενέργειες αυτές θα πρέπει να πραγματοποιούνται έτσι ώστε, μετά την εισαγωγή, ο νεοεισαχθέντας κόμβος να αποτελέσει το δεξιότερο φύλλο και να μην χρειαστούν περαιτέρω ενέργειες (π.χ. swap) μετά την τοποθέτηση του νέου κόμβου στον πίνακα που υλοποιεί το σωρό. Η έκδοση της `HeapInsert()` που θα σχεδιάσετε θα πρέπει να έχει χρονική πολυπλοκότητα $O(\log n)$ και να διατηρεί τις ιδιότητες του σωρού. [20%]

Άσκηση 3 [20 Μονάδες]

Παρουσιάστε έναν αλγόριθμο, ο οποίος, δοθέντων ενός μη-ταξινομημένου πίνακα A που αποθηκεύει n ακέραιους αριθμούς και ενός οποιουδήποτε ακέραιου αριθμού x , καθορίζει αν υπάρχουν δύο στοιχεία του πίνακα που το πηλίκό τους να ισούται με x . Η χρονική πολυπλοκότητα του αλγορίθμου σας θα πρέπει να είναι $O(n \log n)$. [20%]

Υπόδειξη: Θα πρέπει να χρησιμοποιηθούν αποδοτικές τεχνικές ταξινόμησης και εύρεσης στοιχείου σε ταξινομημένο πίνακα.