

HY240: Δομές Δεδομένων
Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2015-16
Παναγιώτα Φατούρου

4^ο Σετ Ασκήσεων

Ημερομηνία Παράδοσης: Κυριακή, 10 Ιανουαρίου 2015

Τρόπος Παράδοσης: Οι ασκήσεις μπορούν να παραδοθούν σε ηλεκτρονική μορφή χρησιμοποιώντας το πρόγραμμα turnin (δείτε οδηγίες στην ιστοσελίδα του μαθήματος). Ασκήσεις που παραδίδονται μετά τις 16:00 της Κυριακής, 10/1/2015 θεωρούνται εκπρόθεσμες. Λύσεις ασκήσεων που αποστέλλονται μέσω ηλεκτρονικού ταχυδρομείου δεν θα γίνονται δεκτές.

Άσκηση 1 [20 μονάδες]

Έστω ότι για κάθε κόκκινο μαύρο δένδρο T διατηρούμε μια μεταβλητή bh στην οποία αποθηκεύουμε το μαύρο ύψος του δένδρου. Για παράδειγμα, το δένδρο αναπαρίσταται ως ένα struct με πεδία ένα δείκτη R στη ρίζα του δένδρου και έναν ακέραιο bh στον οποίο αποθηκεύουμε το μαύρο ύψος του δένδρου.

α. Επιχειρηματολογήστε ότι το μαύρο ύψος bh του T μπορεί να υπολογίζεται σωστά από τις διαδικασίες εισαγωγής ή διαγραφής κόμβου στο T , χωρίς να απαιτείται επιπλέον αποθηκευτικός χώρος στους κόμβους του δένδρου και χωρίς να επηρεάζεται η ασυμπτωτική χρονική πολυπλοκότητα των διαδικασιών αυτών. Επιχειρηματολογήστε επίσης ότι, δεδομένου ότι το bh του T είναι γνωστό, καθώς διατρέχουμε το δένδρο T καθοδικά, από τη ρίζα προς οποιοδήποτε φύλλο, το μαύρο ύψος κάθε κόμβου του μονοπατιού που διασχίζουμε μπορεί να υπολογιστεί σε $O(1)$ χρόνο ανά προσπελαζόμενο κόμβο (έτσι ώστε τα μαύρα ύψη όλων των κόμβων του μονοπατιού να μπορούν να υπολογιστούν σε λογαριθμικό χρόνο). [10%]

β. Έστω ένα κόκκινο-μαύρο δένδρο T με μαύρο ύψος bh και έστω n ο αριθμός των κόμβων του. Έστω k ένας ακέραιος τέτοιος ώστε $bh > k$. Περιγράψτε αλγόριθμο με χρονική πολυπλοκότητα $O(\log n)$, ο οποίος βρίσκει το μαύρο κόμβο με το μεγαλύτερο κλειδί μεταξύ εκείνων με μαύρο ύψος k στο T . [10%]

Άσκηση 2 [40 μονάδες]

α. Παρουσιάστε αλγόριθμο (υπό τη μορφή ψευδοκώδικα) ο οποίος δοθέντος ενός σωρού και ενός δείκτη σε ένα στοιχείο του σωρού, θα αλλάζει την προτεραιότητα του στοιχείου αυτού. Συγκεκριμένα, ζητείται η υλοποίηση μιας συνάρτησης $\text{ChangePriority}(H, j, k)$ η οποία θα αλλάζει την προτεραιότητα του στοιχείου με δείκτη j στο σωρό H έτσι ώστε η νέα προτεραιότητά του να είναι k . Μετά την αλλαγή της προτεραιότητας του στοιχείου j , ο αλγόριθμός σας θα πρέπει να ελέγχει αν έχει καταστρατηγηθεί η ιδιότητα της μερικής διάταξης και να την επαναφέρει, εάν χρειάζεται. Ο αλγόριθμός σας θα πρέπει να έχει χρονική πολυπλοκότητα $O(\log n)$. [20%]

β. **Ξένα σύνολα που υποστηρίζουν τη λειτουργία της ένωσης.** Θεωρήστε την ακόλουθη στρατηγική για μείωση του ύψους ενός πάνω δένδρου.

«Συνενώνουμε το δένδρο με το μικρότερο ύψος στο δένδρο με το μεγαλύτερο ύψος, δηλαδή κάνουμε τη ρίζα του δένδρου με το μικρότερο ύψος να δείχνει στη ρίζα του δένδρου με το μεγαλύτερο ύψος».

Παρουσιάστε ψευδοκώδικα για μια έκδοση της `UpTreeUnion()` η οποία υλοποιεί την παραπάνω στρατηγική. Έστω ότι T είναι ένα δένδρο με δείκτες προς τα πάνω (up tree) το οποίο αναπαριστά ένα σύνολο μεγέθους n που δημιουργήθηκε με τη συνένωση n συνόλων μεγέθους 1 χρησιμοποιώντας την έκδοση της `UpTreeUnion()` που σχεδιάσατε. Μπορείτε να συμπεράνετε κάτι για το ύψος του T ; [20%]

Άσκηση 3 [30 Μονάδες]

- α. Ζητείται να βρεθούν τα 1000 πιο ακριβά στοιχεία μιας μη-ταξινομημένης λίστας τιμών η οποία περιέχει 2^7 στοιχεία. Δύο πιθανά σχήματα για την επίλυση του προβλήματος αυτού είναι τα εξής:

Σχήμα Α: Έκτελεση γραμμικής αναζήτησης στη λίστα 1000 φορές, όπου στην k -οστή αναζήτηση αναζητείται το k -οστό μεγαλύτερο στοιχείο του πίνακα.

Σχήμα Β: Αποθήκευση των στοιχείων της λίστας σε έναν πίνακα, ταξινόμηση του πίνακα και ανάκτηση των 1000 μεγαλύτερων στοιχείων (μετά την ταξινόμηση).

Δίδεται ότι η διάσχιση λίστας 100 στοιχείων πραγματοποιείται σε χρόνο 0.0001 msec και η ταξινόμηση πίνακα 100 στοιχείων πραγματοποιείται σε χρόνο επίσης 0.0001 msec. Ποιο από τα παραπάνω δύο σχήματα είναι το πιο αποδοτικό και γιατί; [15%]

- β. Δίδεται πίνακας μεγέθους N , του οποίου κάθε ένα από τα N στοιχεία, είναι ένα ζεύγος (x,y) που αναπαριστά ένα σημείο στο χώρο με ακέραιες συντεταγμένες x, y . Παρουσιάστε αλγόριθμο που θα βρίσκει και θα διαγράφει από τον πίνακα τα σημεία με τις ίδιες συντεταγμένες. Ο αλγόριθμός σας θα πρέπει να εκτελείται σε χρόνο $O(n \log n)$. [15%]

Παράδειγμα: Θεωρήστε πως ο αρχικός πίνακας είναι 10 θέσεων και περιέχει τα στοιχεία (1,3), (9,11), (6,7), (1,3), (5,6), (9,11), (7,8), (1,3), (9,11), (6,7). Μετά την εκτέλεση του αλγορίθμου σας, ο πίνακας πρέπει να περιέχει έγκυρα στοιχεία στις 5 πρώτες θέσεις του, ενώ στις υπόλοιπες θέσεις του πρέπει να περιέχει την τιμή $\langle -1, -1 \rangle$. Τα 5 έγκυρα στοιχεία του πίνακα είναι τα (1,3), (5,6), (6,7), (7,8), (9,11). Δεν είναι απαραίτητο η σειρά με την οποία εμφανίζονται τα στοιχεία αυτά στις πέντε πρώτες θέσεις του πίνακα να είναι ίδια με αυτή που εμφανίζονταν στον αρχικό πίνακα.