

Δομές Δεδομένων

Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστήμων

Τομέας Μαθηματικών

Δομές Δεδομένων
16 Σεπτεμβρίου 2010

- Να απαντηθούν ΟΛΔ (5) τα θέματα.
- Διάρκεια: 2 ½ ώρες.
- Καλή επιτυχία.

~~Θέμα 1~~

Να εκτελεστεί η μέθοδος Partition (η οποία χρησιμοποιείται από την QuickSort) στο παρακάτω διάνυσμα.

11	4	14	2	19	3	25	1	13	6	33	35	0
----	---	----	---	----	---	----	---	----	---	----	----	---

Να δοθούν τα περιεχόμενα του διανύσματος μετά την κάθε μετακίνηση στοιχείων στο διάνυσμα. Ως «στοιχείο-οδηγός» (pivot) να χρησιμοποιηθεί το πρώτο στοιχείο του διανύσματος («1»).

~~Θέμα 2~~Να σχεδιαστεί δομή δεδομένων η οποία υποστηρίζει το σύνολο ακεραίων στο διάστημα {1...n} καθώς και τις παρακάτω λειτουργίες σε O(1) χρόνο στην χειρότερη περίπτωση:

- Insert(x): Εισάγει το στοιχείο x στο σύνολο
- Delete(x): Διαγράφει το στοιχείο x από το σύνολο
- Member(x): Επιστρέφει true εάν το στοιχείο x ανήκει στο σύνολο, false διαφορετικά.

Να περιγραφεί η υλοποίηση της κάθε λειτουργίας.

~~Θέμα 3~~

Να τροποποιήσετε την δομή δεδομένων «Δυαδικό δένδρο αναζήτησης» όπως θεωρείται αναγκαίο έτσι ώστε να υποστηρίζει έναν αφηρημένο τύπο δεδομένων που υλοποιεί ένα σύνολο ακεραίων και έχει τις παρακάτω λειτουργίες:

- Insert(x): Εισάγει το στοιχείο x στο σύνολο
- Delete(x): Διαγράφει το στοιχείο x από το σύνολο
- Member(x): Επιστρέφει true εάν το στοιχείο x ανήκει στο σύνολο, false διαφορετικά.
- Position(x): Επιστρέφει τον αριθμό των στοιχείων του συνόλου που είναι μικρότερα του x.

Να περιγραφεί η υλοποίηση της κάθε λειτουργίας και να αναλυθεί η πολυπλοκότητα της ως συνάρτηση του ύψους h του δένδρου.~~Θέμα 4~~

Να περιγραφεί αλγόριθμος ο οποίος δέχεται n πραγματικούς αριθμούς και υπολογίζει σε χρόνο O(n log m) τους m μεγαλύτερους από αυτούς. Ο αλγόριθμος σας να κάνει χρήση Αφηρημένων Τύπων Δεδομένων. Να απιστολογηθεί η χρονική πολυπλοκότητα.

~~Θέμα 5~~Να προτείνεται μία υλοποίηση η οποία υποστηρίζει τον αφηρημένο τύπο δεδομένων Min-Max_Priority_Queue, δηλ. μία ουρά προτεραιότητας η οποία ταυτόχρονα υποστηρίζει τις λειτουργίες findMIN, deleteMIN, findMAX, deleteMAX, insert, isEmpty, size. Ζητούμενη είναι O(log n) πολυπλοκότητα χειρότερης περίπτωσης για κάθε λειτουργία. Να περιγράψετε και να αναλύσετε την πολυπλοκότητα κάθε λειτουργίας.