

Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστήμων

Τομέας Μαθηματικών

Δομές Δεδομένων

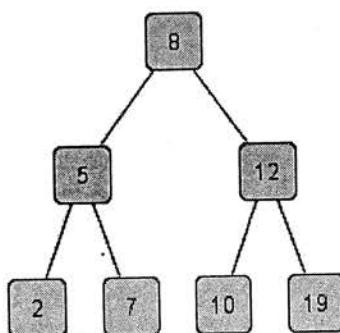
14 Ιουλίου 2011

- Να απαντηθούν **ΟΛΑ (5)** τα θέματα.
- Διάρκεια: $2 \frac{1}{2}$ ώρες.
- Καλή επιτυχία.

Θέμα 1^ο

A. Να περιγραφεί ο τρόπος/αλγόριθμος σύμφωνα με τον οποίο εκτελείται η **διαγραφή** ενός στοιχείου από ένα δυαδικό δένδρο αναζήτησης.

B. Δίνεται το παρακάτω δυαδικό δένδρο αναζήτησης όπου κάθε κόμβος του δένδρου αναγράφει μόνο το κλειδί του στοιχείου που περιέχει.



Με βάση την απάντησή σας στο ερώτημα A, να περιγραφεί το δυαδικό δένδρο αναζήτησης που προκύπτει μετά από κάθε μία από τις παρακάτω διαγραφές των στοιχείων του.

1. delete(8)
2. delete(10)
3. delete(5)
4. delete(7)

Θέμα 2^ο

Σε μία δομή «B-δένδρου τάξεως 3» (ή αλλιώς «2-3 δένδρο») η οποία είναι αρχικά άδεια, εισάγονται στοιχεία τα οποία έχουν ως κλειδιά τους αριθμούς:

<1, 2, 3, 10, 15, 16, 17, 4, 5, 6, 7>

Να περιγραφεί το «B-δένδρου τάξεως 3» που προκύπτει μετά την εισαγωγή **κάθε στοιχείου**. Τα στοιχεία εισάγονται με την σειρά που παρουσιάστηκαν. Κάθε κόμβος του δένδρου αναγράφει μόνο τα κλειδιά των στοιχείων που περιέχει.

Σημείωση: Η απάντηση σας θα πρέπει να περιέχει 11 δένδρα, ένα για την εισαγωγή κάθε στοιχείου.

Θέμα 3^ο

Να υπολογιστεί ο ελάχιστος αριθμός στοιχείων που μπορεί να περιέχει ένα «Β-δένδρου τάξεως 3» το οποίο έχει ύψος h . Να εξηγήσετε αναλυτικά κάθε βήμα του υπολογισμού σας.

Συμβουλή. Για διευκόλυνσή σας, γράψτε πρώτα τον ορισμό του «Β-δένδρου τάξεως m » και προσαρμόστε τον για $m=3$.

Θέμα 4^ο

Σχεδιάστε έναν αλγόριθμο ο οποίος δεχόμενος μία συλλογή m διατεταγμένων, κατά αύξουσα σειρά, λιστών στοιχείων προερχόμενων από ένα ολικά διατεταγμένο σύνολο, επιστρέφει σε αύξουσα σειρά τα k μικρότερα στοιχεία τους (από το σύνολο των στοιχείων όλων των λιστών). Ζητούμενη είναι χρονική πολυπλοκότητα $O(k \log m)$. Ο αλγόριθμος σας να κάνει χρήση Αφηρημένων Τύπων Δεδομένων.

Θέμα 5^ο

Να προτείνεται μία υλοποίηση η οποία στηρίζεται μονό στη δομή δεδομένων του σωρού (μεγίστου ή/και ελαχίστου) και υποστηρίζει τον αφηρημένο τύπο δεδομένων Min-Max_Priority_Queue, δηλ. μία ουρά προτεραιότητας η οποία ταυτόχρονα υποστηρίζει τις λειτουργίες: $findMIN$, $deleteMIN$, $findMAX$, $deleteMAX$, $insert$, $isEmpty$, και $size$. Ζητούμενη είναι $O(\log n)$ πολυπλοκότητα χειρότερης περίπτωσης για κάθε $insert$, $isEmpty$, όπου n είναι ο αριθμός των στοιχείων που βρίσκονται στην ουρά προτεραιότητας. Να λειτουργάψετε και να αναλύσετε την πολυπλοκότητά κάθε λειτουργίας.

Σημείωση: Εάν χρειαστεί να τροποποιηθεί η δομή δεδομένων του σωρού, να περιγράψετε αναλυτικά τις όποιες τροποποιήσεις.