

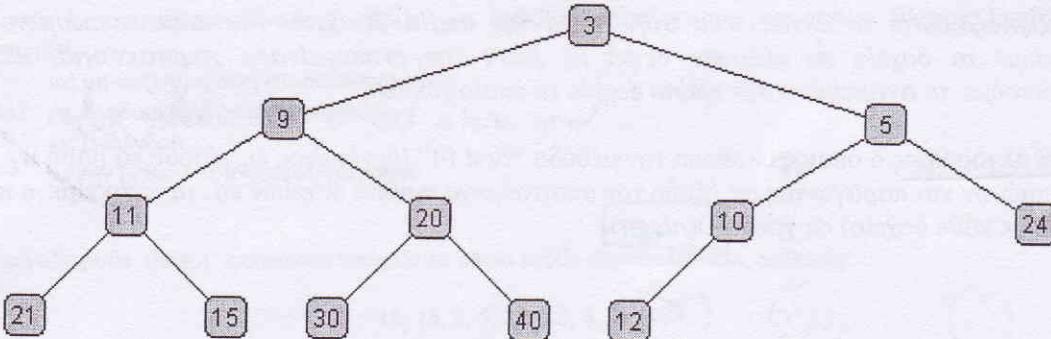
κυνήγος στούρας

Δομές Δεδομένων
22 Σεπτεμβρίου 2008

- Να απαντηθούν **ΟΛΑ (5)** τα θέματα.
- Καλή επιτυχία.

✓ Θέμα 1^o

Δίνεται ο παρακάτω σωρός:



Να περιγραφεί ο σωρός που προκύπτει μετά την εκτέλεση διαδοχικών deleteMin εντολών, (έως ότου αδειάσει). Να δοθεί εικόνα του σωρού μετά από την εκτέλεση κάθε deleteMin εντολής.

heap min

✗ Θέμα 2^o

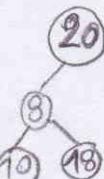
Σχεδιάστε έναν αλγόριθμο, ο οποίος δεχόμενος μία συλλογή k διατεταγμένων, κατά φθίνουσα σειρά, λιστών στοιχείων (προερχόμενων από ένα ολικά διατεταγμένο σύνολο), τις συγχωνεύει σε μία διατεταγμένη λίστα. Ο αλγόριθμος σας να βασίζεται στην χρήση δομών δεδομένων και να έχει χρονική πολυπλοκότητα $O(n \log k)$, οπου n είναι ο συνολικός αριθμός των στοιχείων. Να αιτιολογήσετε την απάντηση σας.

~~mergeSort: Έχω $\lceil \log k \rceil + 1$ έπι. αναδρ. και σε κάθε ένα επιπ. αφιερώνωντες χρόνο $O(1)$ για τη εκδοσή της συγχώνευσης \Rightarrow Συνολικό WC $O(n \log k)$~~

Θέμα 3^o

Σε ένα δυαδικό δένδρο κάθε κόμβος υλοποιείται ως ένα αντικείμενο της κλάσης BNode η οποία παρέχει επίσης τις μεθόδους leftChild() και rightChild() οι οποίες επιστρέφουν το αριστερό και δεξί παιδί, αντίστοιχα (εάν αυτά υπάρχουν, null διαφορετικά) και data() η οποία παρέχει έναν αριθμό ο οποίος είναι αποθηκευμένος στον κόμβο.

- Να γραφεί αναδρομικός ψευτο-κώδικας maxData(v) για τον υπολογισμό του μεγιστού αριθμού που είναι αποθηκευμένος στο υποδένδρο το οποίο έχει τον κόμβο v ως ρίζα. Να αναλυθεί η χρονική πολυπλοκότητα.
- Να γραφεί αναδρομικός ψευτο-κώδικας preorder(v) ο οποίος εκτυπωνει τους κομβους του υποδένδρου το οποίο έχει τον κόμβο v ως ρίζα σε preorder διάταξη. Να αναλυθεί η χρονική πολυπλοκότητα.

✓ Θέμα 4^o

Να αποδειχθεί με επαγωγή ότι ένας σωρός με n κόμβους έχει $\lceil n/2 \rceil$ φύλλα (~~και λογιδιά~~ δεν έχουν λογιδιά)

Θέμα 5^ο

$$W_i \in [0,1]$$

Το πρόβλημα της «συσκευασίας σε δοχεία» (bin packing) είναι το ακόλουθο: Δίνονται η μεταλλικά αντικείμενα, το καθένα με βάρος έως ένα κιλό. Δίνονται επίσης δοχεία (όσα χρειάζονται) στα οποία πρέπει να τοποθετηθούν τα αντικείμενα, έτσι ώστε το βάρος των αντικειμένων που τοποθετούνται σε κάθε δοχείο δεν ξεπερνά το ένα κιλό. Ζητείται να υπολογιστεί ο μικρότερος αριθμός δοχείων που χρειάζεται για να φιλοξενήσει όλα τα αντικείμενα.

Η ευρετική μέθοδος “first fit” για το πρόβλημα της συσκευασίας σε δοχεία είναι η ακόλουθη:

“Επεξεργαζόμαστε τα αντικείμενα στην σειρά την οποία δίνονται. Για κάθε αντικείμενο, ελέγχουμε τα δοχεία σε αύξουσα σειρά με βάση την εναπομένα χωρητικότητα, και τοποθετούμε το αντικείμενο στο πρώτο δοχείο το οποίο χωράει.”

Na σχεδιαστεί αλγόριθμος ο οποίος υλοποιεί την μέθοδο “first fit” (δεχόμενος ως είσοδο τα βάρη w_1, w_2, \dots, w_n , των αντικειμένων και παράγοντας ως έξοδο τον απαιτούμενο αριθμό δοχείων και τα αντικείμενα που τοποθετούνται σε κάθε δοχείο) σε χρόνο $O(n \log n)$.

