

ΘΕΜΑ Γ (18 μονάδες)

Εκτιμώμενος χρόνος επίλυσης: 15 λεπτά

Απαντήστε στις παρακάτω ερωτήσεις. Δεν χρειάζεται να δικαιολογήσετε την απάντησή σας. Αν θέλετε, ακολουθείτε τις ποσότητες απαιτούμενες (απόδεικνύεται να υπάρχει παραπάνω από μία σωστή απάντηση ή ένας και κοινός).

1. Ποιοι από τους παρακάτω ισχυρισμούς είναι σωστοί;
 - a. $n^4 + 3n^3 \log n - 10n^2 + O(n^3 (\log n)^2)$
 - b. $8n^3 + 8n^4 - 30n^2 - 100n^2 + O(n^2)$
 - c. $8n^3 + 8n^4 + 30n^2 + 100n^2 + O(n^2)$
 - d. $4^n = O(n^{100})$
 - e. $\log n \cdot n = O(1)$, όπου $M = O(1)$
 - f. $\log(n!) = O(n \log n)$
 - g. $\log n \cdot n^2 = O(\log n)$
 - h. $2^n = O(n^n)$
2. Οι μέθοδοι ταξινόμησης Insertionsort και Heapsort έχουν
 - a. τις ίδιες απαιτήσεις μνήμης
 - b. την ίδια πολυπλοκότητα χειρότερης περίπτωσης
 - c. την ίδια πολυπλοκότητα μέσης περίπτωσης
3. Οι μέθοδοι ταξινόμησης Quicksort και Mergesort έχουν
 - a. τις ίδιες απαιτήσεις μνήμης
 - b. την ίδια πολυπλοκότητα χειρότερης περίπτωσης
 - c. την ίδια πολυπλοκότητα μέσης περίπτωσης
4. Τα τυχαποιημένα δέντρα δυαδικής αναζήτησης διαφέρουν από τα δέντρα κόκκινου-μαύρου ως προς
 - a. τον χρόνο αναζήτησης στη μέση περίπτωση
 - b. τον χρόνο αναζήτησης στη χειρότερη περίπτωση
 - c. το ύψος του δέντρου στη χειρότερη περίπτωση από την εισαγωγή n στοιχείων
5. Έστω ένα δέντρο 2-3-4 και ένα AVL δέντρο με τον ίδιο αριθμό αντικείμενων. Έστω h το ύψος του δέντρου 2-3-4. Το ύψος του AVL δέντρου είναι πάντα
 - a. τουλάχιστον h
 - b. τουλάχιστον $2h$
 - c. μικρότερο ή ίσο από $2h$
6. Έστω ένας σωρός, στον οποίο έχουν ήδη εισαχθεί 11 αντικείμενα (με τη ρίζα να είναι στο επίπεδο 0 του δέντρου). Το επόμενο αντικείμενο που θα εισαχθεί με την κλήση της insert, θα τοποθετηθεί αρχικά στο επίπεδο
 - a. 2
 - b. 3
 - c. 4

ΘΕΜΑ 2 (14 μονάδες)

Εκτιμώμενος χρόνος επίλυσης: 15-20 λεπτά

Δίνεται ο ακόλουθος ορισμός των κόμβων ενός δέντρου δυαδικής αναζήτησης. Κάθε κόμβος περιέχει ένα ακέραιο κλειδί *key* και δύο συνδέσμους προς το αριστερό και δεξιό υποδέντρο.

```
class Node {  
    int key;  
    Node left, right;  
}
```

(α) [10 μονάδες] Συμπληρώστε τον κώδικα που λείπει παρακάτω έτσι ώστε η μέθοδος `sort(Node h)`, να τυπώνει σε αύξουσα σειρά τα κλειδιά ενός δέντρου δυαδικής αναζήτησης με ρίζα τον κόμβο *h*. Δεν είναι απαραίτητο να συμπληρώσετε όλα τα κενά στον παρακάτω κώδικα για να είναι σωστή η υλοποίηση. Δεν μπορείτε να ορίσετε άλλες μεθόδους, θα πρέπει απλά να συμπληρώσετε ό,τι λείπει στον κώδικα της μεθόδου και να προσθέσετε το πολύ μια εντολή σε κάθε κενή γραμμή.

```
void sort(Node h)  
{  
    if (h == null) .....;  
    .....  
    .....  
    System.out.println(.....);  
    .....  
    .....  
}
```

(β) [4 μονάδες] Θεωρήστε την παρακάτω μέθοδο, η οποία δέχεται ως είσοδο έναν κόμβο h , και ως υποθέσουμε ότι ο h είναι η ρίζα κάποιου δυαδικού δέντρου.

```
void foo(Node h) {  
    if (h==null) System.out.println("Last line I print");  
    else {  
        System.out.println("I am printing one more line");  
        foo(h.left);  
    }  
}
```

Έστω ότι το δέντρο περιέχει N μη κενούς κόμβους. Η μέθοδος αυτή τυπώνει στην έξοδο κάποιες γραμμές, ο αριθμός των οποίων εξαρτάται από τη δομή του δέντρου.

Αν το δέντρο είναι ισοζυγισμένο, πόσες γραμμές τυπώνει η μέθοδος (ως συνάρτηση του N , σε τάξη μεγέθους, με συμβολισμό μεγάλου O);

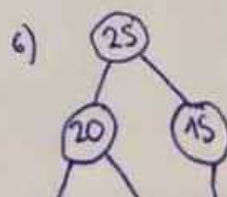
ΘΕΜΑ 3 (28 μονάδες)

Εκτιμώμενος χρόνος επίλυσης: 25-30 λεπτά

(α) [9 μονάδες] Ξεκινώντας από έναν άδικο δυαδικό σωρό, να κάνετε εισαγωγή των εξής στοιχείων (με τη σειρά που δίνονται): 8, 20, 18, 15, 9, 25, 22, 25, 30. Να σχεδιάσετε το δέντρο που προκύπτει, τη στιγμή που έχει ολοκληρωθεί η εκάστοτε εισαγωγή (συνολικά πρέπει να σχεδιάσετε 9 δέντρα). Αν σας διευκολύνει στην παρουσίαση, μπορείτε να σχεδιάσετε προαιρετικά και τα ενδιάμεσα δέντρα που προκύπτουν μετά από κάθε ανάδυση, κατά τη διάρκεια των εισαγωγών.

(β) [6 μονάδες] Έστω ότι έχετε ένα σωρό με τις εξής ιδιότητες: ο σωρός έχει 9 αντικείμενα με ακέραια κλειδιά, και η ρίζα έχει το κλειδί 20. Στο αριστερό υποδέντρο του σωρού ισχύει επίσης ότι: i) κάθε παιδί ενός κόμβου του αριστερού υποδέντρου έχει τιμή το πολύ ίση με το μισό της τιμής του πατέρα του, ii) το δεξιό παιδί έχει τιμή μεγαλύτερη από το αριστερό. Για το δεξιό υποδέντρο του σωρού δεν υπάρχει κάποιος περιορισμός εκτός του ότι πρέπει να ικανοποιείται η ιδιότητα του σωρού. Να σχεδιάσετε ένα δυαδικό δέντρο το οποίο αναπαριστά ένα σωρό με όλες τις παραπάνω ιδιότητες.

(γ) [10 μονάδες] Ξεκινώντας από ένα κενό δέντρο δυαδικής αναζήτησης (ΔΔΑ), εκτελέστε εισαγωγή των εξής 8 κλειδιών, με τη σειρά που δίνονται: 20, 15, 18, 22, 25, 30, 10, 12. Εφαρμόστε την απλή εισαγωγή ως φύλλο για τα πρώτα 7 κλειδιά, και τον αλγόριθμο εισαγωγής στη ρίζα για το τελευταίο κλειδί (αφότου δηλαδή ολοκληρωθούν οι πρώτες 7 εισαγωγές). Σχεδιάστε τα δέντρα που προκύπτουν μετά την ολοκλήρωση της κάθε εισαγωγής, και δείξτε επίσης όλες τις απαιτούμενες περιστροφές που χρειάστηκαν για την εισαγωγή του τελευταίου κλειδιού μέχρι να φτάσει στη ρίζα.



(22 μονάδες)
Εκτιμώμενος χρόνος επίλυσης: 20-25 λεπτά

- (α) [10 μονάδες] Θέλουμε να αποθηκεύσουμε 6000 αντικείμενα σε πίνακα κατακερματισμού. Έστω ότι υπάρχει αυστηρός περιορισμός ότι το μέσο ελάχιστο διαμερίσμων για τις αναζητήσεις δεν πρέπει να υπερβαίνει το 25. Ποιο πρέπει να είναι το ελάχιστο μέγεθος του πίνακα;
- αν χρησιμοποιήσουμε απλή γραμμική διαμερίσμων;
 - αν χρησιμοποιήσουμε διπλό κατακερματισμό;
- Τι από τα 2 μας συμφέρει περισσότερο, με κριτήριο τις απαιτήσεις μνήμης του κάθε συστήματος; Το μέσο ελάχιστο διαμερίσμων στη γραμμική διαμερίσμων είναι $(1 + 1/(1-a^2))/2$ για αναζητήσεις, όπου a ο συντελεστής φορτίου. Αντίστοιχα, στο διπλό κατακερματισμό είναι $1/(1-a)$.

(β) [12 μονάδες] Στο παρακάτω δέντρο 2-3-4, να εισάγετε με αναδρομική εισαγωγή πρώτα το κλειδί 34 και μετά το κλειδί 9. Για κάθε μια από τις 2 εισαγωγές, δείξτε τα βήματα που γίνονται καθώς και το δέντρο που προκύπτει όταν ολοκληρώνεται η κάθε εισαγωγή. Σχεδιάστε επίσης και ένα ισοδύναμο δέντρο κόκκινου-μαύρου για το τελικό δέντρο που προέκυψε μετά την ολοκλήρωση των εισαγωγών.

