



ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

27.01.2011

Δομές Δεδομένων

Εξεταστική Ιανουαρίου 2011

Διδάσκων : Ευάγγελος Μαρκάκης

ΟΝΟΜΑΤΕΠΩΝΥΜΟ:	
ΑΜ:	

Διάρκεια εξέτασης : 2 ώρες και 30 λεπτά

Απαντήστε σε όλα τα θέματα

Θέμα	Βαθμός
1	
2	
3	
4	
5	
Σύνολο:	

ΘΕΜΑ 1 (12 μονάδες)

Απαντήστε με **σωστό** ή **λάθος** για καθεμία από τις παρακάτω προτάσεις. Δεν χρειάζεται να δικαιολογήσετε την απάντησή σας.

1. $n^5 + 160n^2 + n \log n = O(n^3)$
2. $\log_8 n^5 = O(\log_4 n)$
3. $n! = O(n^{10})$
4. $4n^4 - n^3 - 100n^2 = O(n^4)$
5. $f(n) = O(n \log n)$, όπου $f(n) = f(n-1) + n$, για $n \geq 2$, και $f(1) = 1$
6. Σε ένα τυχαιοποιημένο δέντρο δυαδικής αναζήτησης με N κόμβους, ο χρόνος αναζήτησης στη χειρότερη περίπτωση είναι $O(\log N)$.

ΘΕΜΑ 2 (15 μονάδες)

Δίνεται ο ακόλουθος ορισμός μίας ουράς προτεραιότητας με ακέραια κλειδιά που θέλουμε να υλοποιήσουμε με χρήση ταξινομημένου πίνακα αντί για χρήση σωρού.

```
class PQ{
    private int[] pq;
    private int N; //δηλώνει αριθμό ενεργών θέσεων
    PQ(int maxN){ pq = new int[maxN]; N=0; }
    void insert(int v) {...}
    int getmax() {...}
}
```

Να γράψετε σε Java τις μεθόδους `insert` και `getmax`, που ορίζονται παραπάνω, έτσι ώστε ο πίνακας `pq` να διατηρείται πάντα ταξινομημένος. Πιο συγκεκριμένα, η μέθοδος `insert` θα πρέπει να εισάγει ένα στοιχείο και να εξασφαλίζει ότι ο πίνακας θα παραμένει ταξινομημένος (μπορείτε να επιλέξετε αν θα είναι σε αύξουσα ή σε φθίνουσα σειρά). Η μέθοδος `getmax` θα πρέπει να εξάγει τον μέγιστο ακέραιο από την ουρά προτεραιότητας. Τα κλειδιά θα πρέπει πάντα να είναι αποθηκευμένα στις θέσεις 0 έως N-1 (μην ξεχνάτε να ενημερώνετε το πεδίο N στην εισαγωγή και εξαγωγή). Ποια από τις 2 μεθόδους έχει καλύτερη πολυπλοκότητα και γιατί;

ΘΕΜΑ 3 (15 μονάδες)

Δίνεται ο ακόλουθος ορισμός ενός δυαδικού σωρού και κάποιων βασικών λειτουργιών του όπως τις είδαμε και στο μάθημα:

```
class PQ {
    int[] pq; int N;
    PQ(int maxN) { pq = new int[maxN+1]; N = 0; }
    boolean less(int i, int j) {
        return (pq[i] < pq[j]); }
    void exch(int i, int j) {
        int t = pq[i]; pq[i] = pq[j]; pq[j] = t; }
    void swim(int k) {
        while (k > 1 && less(k/2, k)) {
            exch(k, k/2); k = k/2; }
    void insert(int v) {
        pq[++N] = v; swim(N); }
}
```

Έστω ότι ο σωρός αρχικά έχει την ακόλουθη μορφή (το στοιχείο στη θέση 0 του πίνακα pq δεν χρησιμοποιείται):

A) Σχεδιάστε την

	15	10	11	4	9
--	----	----	----	---	---

 αναπαράσταση του σωρού ως πλήρες δυαδικό δέντρο.

B) Έστω ότι εισάγουμε διαδοχικά τα στοιχεία 13, 12, 18, 16, σύμφωνα με τη μέθοδο `insert`, η οποία αρχικά εισάγει ένα στοιχείο στο τέλος και στη συνέχεια καλεί την `swim` για να αποκαταστήσει την ιδιότητα του σωρού. Να σχεδιάσετε τον πίνακα και το αντίστοιχο πλήρες δυαδικό δέντρο που προκύπτει για τον σωρό όταν ολοκληρώνεται κάθε κλήση της `insert`.

ΘΕΜΑ 4 (8 μονάδες)

Δίνεται ο ακόλουθος ορισμός των κόμβων ενός δυαδικού δέντρου.

```
class Node {  
    int key;  
    Node l, r;  
    Node(int x) {key = x; l = r = null; }}
```

Συμπληρώστε τον κώδικα που λείπει παρακάτω έτσι ώστε η μέθοδος `traverse(Node h)`, να κάνει μεταδιατεταγμένη (postorder) διάσχιση σε ένα δέντρο ξεκινώντας από τον κόμβο `h` και όταν επισκέπτεται έναν κόμβο, να μεταβάλλει την τιμή του κλειδιού σύμφωνα με τη συνάρτηση $h(k) = k + (2k \bmod 3)$, και να τυπώνει τη νέα τιμή. Δεν είναι απαραίτητο να συμπληρώσετε όλα τα κενά στον παρακάτω κώδικα για να είναι σωστή η υλοποίηση.

```
void traverse(Node h)  
{  
    if (.....) return;  
  
    .....  
  
    .....  
  
    h.key = .....;  
  
    System.out.println(.....);  
  
    .....  
  
    .....  
}
```

ΘΕΜΑ 5 (30 μονάδες)

Απαντήστε στις παρακάτω ερωτήσεις δικαιολογώντας την απάντησή σας.

1. [10] Θέλετε να κάνετε αναζήτηση και εισαγωγή κλειδιών σε μία δομή και θέλετε ο μέσος χρόνος για τις λειτουργίες αυτές να είναι $O(\log N)$. Εξηγήστε ποια από τις παρακάτω δομές δεν είναι κατάλληλη:

- i) Δέντρα δυαδικής αναζήτησης
- ii) Τυχαιοποιημένα δέντρα δυαδικής αναζήτησης
- iii) Ταξινομημένος πίνακας
- iv) Αταξινόμητη λίστα
- v) Δέντρα κόκκινου-μαύρου
- vi) Κατακερματισμός με χωριστή αλυσίδωση

2. **[5]** Γιατί στον κατακερματισμό με χωριστή αλυσίδωση χρησιμοποιούμε συνήθως μη διατεταγμένες λίστες;

3. **[10]** Επιλέξτε 10 ακραίους αριθμούς στο διάστημα $[1, 100]$. Έστω ότι θέλετε να εφαρμόσετε κατακερματισμό με χωριστή αλυσίδωση και επιλέγετε ως συνάρτηση κατακερματισμού την $h(k) = k \bmod 7$. Σχεδιάστε τη μορφή του πίνακα όταν εισαχθούν διαδοχικά όλα τα στοιχεία.

4. [5] Έστω ότι θέλετε να εισάγετε το κλειδί 41 στο παρακάτω δέντρο 2-3-4. Να σχεδιάσετε το νέο δέντρο 2-3-4 μετά την εισαγωγή καθώς και το δέντρο κόκκινου-μαύρου στο οποίο αντιστοιχεί.



