



ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ &  
ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

Τμήμα Εφαρμοσμένων Μαθηματικών  
Εαρινό εξάμηνο '07  
ΕΜ 202, Θεωρία Αλγορίθμων, Π. Λεκέας  
2 Ιουλίου 2007

### Τελική Εξέταση

Έχετε στη διάθεσή σας 3 ώρες για να λύσετε τα προβλήματα. Γράψτε τις απαντήσεις σας στο χώρο που σας δίνεται. Χρησιμοποιείτε για πρόχειρο τις τελευταίες σελίδες. Να δικαιολογείτε σε κάθε περίπτωση την απάντησή σας. Το άριστα για αυτό το διαγώνισμα είναι 80 μονάδες.

Όνοματεπώνυμο

A.M.

Βαθμοί

1.

2.

3.

4.

5.

6.

Σύνολο \_\_\_\_\_

1. (15 μονάδες)
- α) Λύστε την αναδρομική εξίσωση:  $T(n)=9T(n/27)+(\sqrt[3]{n})^2$
  - β) Λύστε την αναδρομική εξίσωση:  $T(n)=7T(n/2)+n^3$
  - γ) Λύστε την αναδρομική εξίσωση:  $T(n)=T(n/7)+1$

2. Το πρόβλημα της δυαδικής αναζήτησης (Binary Search) είναι το εξής: «Σε ταξινομημένο πίνακα A βρες το στοιχείο κ». (6 μονάδες) α) Προτείνετε ένα αλγόριθμο τύπου Διαίρει και Κυρίευε (Divide and Conquer) που λύνει το πρόβλημα της δυαδικής αναζήτησης. (9 μονάδες) β) Γράψτε την αναδρομική εξίσωση που προκύπτει και λύστε την.

3. (6 μονάδες) α) Σε τί μας μας χρησιμεύει ο αλγόριθμος MAX-HEAPIFY; (σας δίνεται στο παράρτημα). Ποιός είναι ο χρόνος εκτέλεσης χειρότερης περίπτωσης του; Γιατί;
- (9 μονάδες) β) Διαθέτουμε  $n$  ακέραιους στο διάστημα  $0$  έως  $n^2-1$ . Μπορώ να τους ταξινομήσω σε χρόνο  $O(n)$ ; Δικαιολογείστε την απάντησή σας.

4. (15 μονάδες) Σας δίνεται στο παράρτημα ο αλγόριθμος ΑΡΧΕΤΥΠΙΚΟ-ΕΣΔ ( $G, w$ ) [ή Generic-MST ( $G, w$ )] που χρησιμοποιήσαμε στο μάθημα για να κάνουμε επέκταση κατά μία ακμή κάθε φορά ενός ελαφρύτατου συνεκτικού δένδρου. Προτείνετε μία αναλλοίωτη συνθήκη και με τη βοήθειά της αποδείξτε την ορθότητα του πιο πάνω αλγορίθμου.

5. (5 μονάδες) α) Βρείτε ένα ελαφρύτατο συνεκτικό δένδρο του γράφου A του παραρτήματος χρησιμοποιώντας τον αλγόριθμο του Prim MST-PRIM  $(G, w, r)$  που σας δίνεται στο παράρτημα. Σε ένα συνοπτικό πίνακα σημειώστε την πορεία εκτέλεσης του αλγορίθμου καθώς υπολογίζει το ελαφρύτατο συνεκτικό δένδρο με ρίζα τον κόμβο A.
- (5 μονάδες) α) Υπολογίστε για το γράφο B του παραρτήματος ένα ελαφρύτατο συνεκτικό δένδρο χρησιμοποιώντας τον αλγόριθμο του Kruskal.

6. (10 μονάδες) Υποθέστε ότι σε ένα υπολογιστικό σύστημα έχουμε προς εκτέλεση  $n$  διαφορετικά task. Αν το κάθε task χρειάζεται  $w_1, \dots, w_n$  χρονικές στιγμές για να εκτελεστεί από τη CPU, και ο βαθμός σημαντικότητας του κάθε task είναι  $v_1, \dots, v_n$  αντίστοιχα, τότε ποιός είναι ο πιο συμφέρον συνδυασμός με τον οποίο πρέπει να εκτελεστούν τα task αν η CPU διαθέτει  $W$  χρονικές στιγμές για αυτή τη δουλειά; Δώστε έναν αλγόριθμο που υπολογίζει αυτούς τους συνδυασμούς για την περίπτωση που υποθέτουμε ότι υπάρχουν απεριόριστα task από κάθε είδος και ότι μας επιτρέπεται να εκτελέσουμε περισσότερα από ένα task του ίδιου είδους στη CPU.

## ΠΑΡΑΡΤΗΜΑ

GENERIC-MST( $G, w$ )

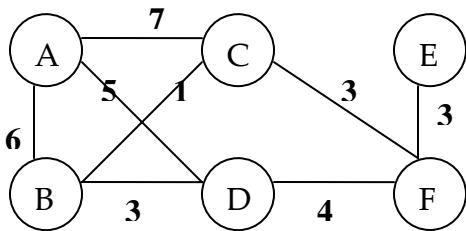
```
1  $A \leftarrow \emptyset$ 
2 while  $A$  does not form a spanning tree
3   do find an edge  $(u, v)$  that is safe for  $A$ 
4      $A \leftarrow A \cup \{(u, v)\}$ 
5 return  $A$ 
```

---

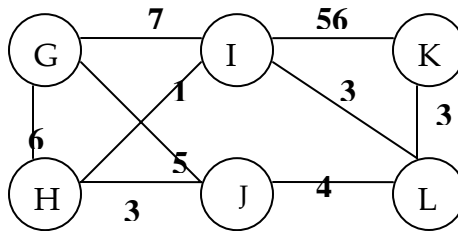
MST-PRIM( $G, w, r$ )

```
1 for each  $u \in V [G]$ 
2   do  $key[u] \leftarrow \infty$ 
3      $\pi[u] \leftarrow NIL$ 
4  $key[r] \leftarrow 0$ 
5  $Q \leftarrow V [G]$ 
6 while  $Q \neq \emptyset$ 
7   do  $u \leftarrow EXTRACT-MIN(Q)$ 
8     for each  $v \in Adj[u]$ 
9       do if  $v \in Q$  and  $w(u, v) < key[v]$ 
10         then  $\pi[v] \leftarrow u$ 
11            $key[v] \leftarrow w(u, v)$ 
```

---



Γράφος A



Γράφος B

MAX-HEAPIFY( $A, i$ )

```
1  $l \leftarrow LEFT(i)$ 
2  $r \leftarrow RIGHT(i)$ 
3 if  $l \leq heap-size[A]$  and  $A[l] > A[i]$ 
4   then  $largest \leftarrow l$ 
5   else  $largest \leftarrow i$ 
6 if  $r \leq heap-size[A]$  and  $A[r] > A[largest]$ 
7   then  $largest \leftarrow r$ 
8 if  $largest \neq i$ 
9   then exchange  $A[i] \leftrightarrow A[largest]$ 
10   MAX-HEAPIFY( $A, largest$ )
```

---

# ΠΡΟΧΕΙΡΟ