ΕΠΛ 231: Δομές Δεδομένων και Αλγόριθμοι

Κατ’οίκον Εργασία 1

Ημερομηνία Παράδοσης: 29/09/10

1. [12 μονάδες] Έξι αλγόριθμοι έχουν φτιαχτεί για το ίδιο πρόβλημα με τους πιο κάτω χρόνους εκτέλεσης. (Υποθέστε ότι οι συναρτήσεις υπολογίζουν με ακρίβεια τον αριθμό των βασικών πράξεων που εκτελούν οι αλγόριθμοι σε δεδομένα μεγέθους n.) Πόσο θα μεγαλώσει ο χρόνος εκτέλεσης του κάθε αλγόριθμου αν (i) διπλασιάσουμε το μέγεθος του δεδομένου εισόδου, ή (ii) αυξήσουμε το μέγεθος του δεδομένου εισόδου κατά 1;

   (α) n^2

   (β) n^3

   (γ) $100\, n^2$

   (δ) $n \lg n$

   (ε) 2^n

   (ζ) 2^{2^n}

2. [30 μονάδες] Να τοποθετήσετε τις πιο κάτω συναρτήσεις σε αύξουσα σειρά ως προς την τάξη τους *αποδεικνύοντας* την απάντησή σας. Συγκεκριμένα, σε περίπτωση που προτείνετε κάποια σειρά, $g_1(n)$, $g_2(n)$, $g_3(n)$, να αποδείξετε ότι κάθε συνάρτηση ανήκει στην τάξη Ο της επόμενης ενώ δεν ανήκει στην τάξη Ο της προηγούμενης.

   $f_1(n) = 4\sqrt{n}$

   $f_2(n) = n^3 - 3 \cdot n^{2.5}$

   $f_3(n) = n^{2.5}$

3. [24 μονάδες] Να αποφασίσετε ποιες από τις πιο κάτω προτάσεις είναι αληθείς δίνοντας είτε απόδειξη είτε κάποιο αντιπαράδειγμα.

   (i) Αν $f(n) \in \Theta(f'(n))$ και $g(n) \in \Theta(g'(n))$ τότε $f(n) + g(n) \in \Theta(\min(f'(n),\ g'(n)))$.

   (ii) Για κάθε συνάρτηση $f(n)$ και $g(n)$ είτε $f(n) \in \Omega(g(n))$ είτε $g(n) \in \Omega(f(n))$.

   (iii) $\sum_{i=1}^{n} i^k \in O(n^{k+1})$

4. [16 μονάδες] Να αναλύσετε τον χρόνο εκτέλεσης χειρίστης περίπτωσης των πιο κάτω σαν συνάρτηση του n ως προς την τάξη Θ.

   Α. [6 μονάδες]

```
selectionsort(int A[], int n){
     for (int i=1; i<n; i++)
          k=i;
          for (j = i+1; j ≤ n; j++)
               if A[j] < A[k];
                    swap(A[i],A[k]);
}
```

B. [10 μονάδες]

```
sum = 0;
for (i=2n; i≥1; i=i/2)
    for (j = 1; j ≤ i/2; j++)
         for ( k = n; k ≥ 1; k--)
               sum++;
```

C. [προαιρετική]

```
int sum = 0;

for (i = 1; i ≤ √n; i++)
   for (j = 1; j ≤ √n; j=j*2)
      for (k = 1; k ≤ j; k++)
           sum++;

int sum=0;
for ( int i=1; i ≤ n; i++)
     if (i mod 3 == 0)
          for ( int j=1; j ≤ i; j++)
               sum++;
```

5. [18 μονάδες] Να υπολογίσετε τον χρόνο εκτέλεσης της παρακάτω αναδρομικής διαδικασίας λύνοντας οποιαδήποτε αναδρομική εξίσωση συναντήσετε.

```
recursive(int n){
   int sum=0;
   if  (n ≤ 1) return;
   else{
     for (int i=1; i ≤ n; i++){
        if (i mod 3 == 0)
           for ( int j=1; j ≤ i; j++)
               sum++;
     }
     recursive(n-1);
   }
}
```