

	Πανεπιστήμιο Θεσσαλίας Τμήμα Πληροφορικής με Εφαρμογές στη Βιοϊατρική
	Δομές Δεδομένων και Αλγόριθμοι Ακαδημαϊκό Έτος 2013-2014 http://users.ucg.gr/~emarkou/greek/teach/id-5 Ε. Μάρκου

1ο Σετ Ασκήσεων

Ημερομηνία Παράδοσης: Παρασκευή 11 Απριλίου 2014.

Τρόπος Παράδοσης: Χειρόγραφες ή εκτυπωμένες σελίδες που τις αφήνετε στη γραμματεία του τμήματος (κατά τις ώρες λειτουργίας της).

Άσκηση 1 [30 μονάδες]

Θεωρήστε την ακόλουθη διαδικασία, όπου η `swap(int a, int b)` ανταλλάσσει την τιμή της μεταβλητής `a` με την τιμή της μεταβλητής `b`.

```
void MySort(table T[0..n-1]: array of integers)
{
    for m = 0 to n-2 do
    {
        j = m;
        for k = m+1 to n-1 do
        {
            if T[k] < T[j] then j = k;
        }
        swap(T[m], T[j]);
    }
}
```

α. Παρουσιάστε σύντομη περιγραφή του τρόπου λειτουργίας της `MySort()`. Ποιο είναι το αποτέλεσμα της `MySort()`; [7 μονάδες]

β. Εκτελέστε με το χέρι τη `MySort(T[0..5])` για την περίπτωση που `T = [50, 20, 30, 10, 3, 15]`. [10 μονάδες]

γ. Ποιά είναι η χρονική πολυπλοκότητα της `MySort()`; Βρείτε την τάξη στην οποία ανήκει (συμβολισμός Θ). [13 μονάδες]

Άσκηση 2 [40 μονάδες]

Θεωρήστε τον ακόλουθο αναδρομικό αλγόριθμο, ο οποίος υπολογίζει το ελάχιστο στοιχείο ενός μη-ταξινομημένου πίνακα ακεραίων $T[a..b]$ με $n = b-a+1 = 2^k$ ($k > 0$) στοιχεία, όπου a, b είναι θετικοί ακέραιοι που καθορίζουν τη θέση του πρώτου και του τελευταίου στοιχείου του πίνακα.

```
int FindMin(table T: array of int, int a, int b)
/* επιστρέφει το ελάχιστο στοιχείο του μη-ταξινομημένου πίνακα T */
{
    int middle;

    if (b-a == 0) return T[a];
    middle = κάτω_ακέραιο_μέρος{(a+b)/2};
    for j=a to middle do
        if (T[j] > T[j+middle]) then swap(T[j],T[j+middle]);
    min = FindMin(T, a, middle);
    return min;
}
```

α. Εκτελέστε με το χέρι την $\text{FindMin}(T, 1, 8)$ για την περίπτωση που $T = [8, 3, 25, 32, 15, 7, 20, 1]$. Παρουσιάστε σύντομη περιγραφή του τρόπου λειτουργίας του αλγορίθμου. [5 μονάδες]

β. Παρουσιάστε αναδρομική σχέση που να περιγράφει τη χρονική πολυπλοκότητα της $\text{FindMin}()$. Υπολογίστε την τάξη πολυπλοκότητας της $\text{FindMin}()$. [10 μονάδες]

γ. Δίνεται ο ακόλουθος αναδρομικός αλγόριθμος ($\text{FindMinMax}()$) που υπολογίζει και το ελάχιστο και το μέγιστο στοιχείο του πίνακα T . Μετά την εκτέλεση του αλγορίθμου το ελάχιστο στοιχείο του πίνακα βρίσκεται στη θέση $T[a]$ ενώ το μέγιστο στοιχείο βρίσκεται στη θέση $T[b]$.

```
void FindMinMax(table T, int a, int b)
{
    int middle;

    if (b-a==0) return;
    middle = κάτω_ακέραιο_μέρος{(a+b)/2};
    for j=a to middle do
        if (T[j] > T[j+middle-a+1]) then swap(T[j],T[j+middle-a+1]);
    FindMinMax(T, a, middle);
    FindMinMax(T, middle+1,b);
}
```

i. Εκτελέστε με το χέρι την FindMinMax(T,1,8) για την περίπτωση που $T=[8,3,25,32,15,7,20,1]$. Παρουσιάστε σύντομη περιγραφή του τρόπου λειτουργίας του αλγορίθμου. [12 μονάδες]

ii. Παρουσιάστε αναδρομική σχέση $T(n)$ που να περιγράφει τη χρονική πολυπλοκότητα της FindMinMax(). Τι τάξης είναι η πολυπλοκότητα της FindMinMax() και γιατί; [13 μονάδες]

Άσκηση 3 [30 μονάδες]

α. Απαντήστε τα ακόλουθα ερωτήματα. [12 μονάδες]

i. Να αποδείξετε ότι $n^2 + 5n - 10 = \Theta(n^2)$.

ii. Ισχύει πως $n^2 + 5n - 10 = O(n^3 \log n)$ και γιατί;

iii. Ισχύει πως $\sqrt{n^5} \log(\sqrt{n^5}) = O(n^3)$ και γιατί;

iv. Αποδείξτε ότι αν $f(n) = O(g(n))$ και $g(n) = O(h(n))$ τότε $f(n) = O(h(n))$.

v. Αποδείξτε ότι αν $f(n) = O(g(n))$ τότε $g(n) = \Omega(f(n))$.

β. Βρείτε την χρονική πολυπλοκότητα $T(n)$ των παρακάτω αλγορίθμων, όπου \sqrt{n} είναι η τετραγωνική ρίζα του n . [18 μονάδες]

```
i. procedure Peculiar(int n)
{
    for j = 1 to sqrt(n) do
        for k = 1 to n do
            if (k mod 2 == 0) then x = x+1;
}
```

```
ii. procedure Puzzle(int n)
{
    for j = 1 to n2 do
        for k=j to n do
            for m = sqrt(n)+1 to 2*sqrt(n) do x = x+1;
}
```

```
iii. procedure Mystery(int n)
{
    for i = 1 to sqrt(n) do
        for j=sqrt(n)+1 to n do
            for k = n-10 to n do x = x+1;
}
```