

## Η/Υ 432: Δομές Δεδομένων

### Χειμερινό Εξάμηνο Ακαδημαϊκού Έτους 2006-2007

### Παναγιώτα Φατούρου – Ευριπίδης Μάρκου

### 1<sup>ο</sup> Σετ Ασκήσεων

**Ημερομηνία Παράδοσης:** Παρασκευή, 9/11, ώρα: 12:00-14:00.

**Τρόπος Παράδοσης:** Οι ασκήσεις παραδίδονται στους βοηθούς του μαθήματος στο γραφείο Α25 το αργότερο μέχρι την ημερομηνία παράδοσης. Το γραφείο των βοηθών του μαθήματος είναι το Α25.

#### Άσκηση 1 [30 μονάδες]

Θεωρήστε την ακόλουθη διαδικασία, όπου η `swar(int a, int b)` ανταλλάζει 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;
        }
        swar(T[m], T[j]);
    }
}
```

- α. Παρουσιάστε σύντομη περιγραφή του τρόπου λειτουργίας της `MySort()`. Ποιο είναι το αποτέλεσμα της `MySort()`; [7%]
- β. Ιχνηλατίστε (δηλαδή «εκτελέστε» στο χαρτί - κάνετε trace) τη `MySort(T[0..5])` για την περίπτωση που `T = [50, 20, 30, 10, 3, 15]`. [10%]
- γ. Μελετήστε την τάξη της χρονικής πολυπλοκότητας της `MySort()`. [13%]

**Υπόδειξη:** Αφού βρείτε τη χρονική πολυπλοκότητα  $T(n)$  του αλγορίθμου, βρείτε την τάξη στην οποία ανήκει (συμβολισμός  $\Theta$ ).

#### 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 το πρώτο και το τελευταίο στοιχείο 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. Ιχνηλατίστε την 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%]

β. Παρουσιάστε αναδρομική σχέση που να περιγράφει τη χρονική πολυπλοκότητα της FindMin(). Τι τάξης είναι η πολυπλοκότητα της FindMin() και γιατί; [10%]

γ. Δίνεται ο ακόλουθος αναδρομικός αλγόριθμος (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]. [25%]

```

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. Ιχνηλατίστε 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%]

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

### Άσκηση 3 [30 μονάδες]

α. Απαντήστε τα ακόλουθα ερωτήματα. [12%]

- i. Ισχύει πως  $n^2 + 5n - 10 = \Theta(n^2)$  και γιατί;
- 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 τη μεταβατική ιδιότητα για το O και την αντίστροφη συμμετρική ιδιότητα των O, Ω.

γ. Βρείτε την τάξη της χρονικής πολυπλοκότητας T(n) των παρακάτω αλγορίθμων, όπου  $\sqrt{n}$  είναι η τετραγωνική ρίζα του n. [18%]

```

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 n do  
        for k = j to n2 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;  
    }
```

**Σημείωση:** Υπολογίστε τη χρονική πολυπλοκότητα κάθε συνάρτησης και αναφέρετε τι τάξης είναι. Δεν χρειάζεται να αποδείξετε βάσει των ορισμών των  $O$ ,  $\Omega$  και  $\Theta$  ότι η πολυπλοκότητα που υπολογίσατε είναι της τάξης που ισχυρίζεστε.