

HY240: Δομές Δεδομένων
Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2017-18
Παναγιώτα Φατούρου

1^η Σειρά Ασκήσεων

Ημερομηνία Παράδοσης: Δευτέρα, 16 Οκτωβρίου 2017

Τρόπος Παράδοσης: Οι ασκήσεις μπορούν να παραδοθούν είτε σε ηλεκτρονική μορφή χρησιμοποιώντας το πρόγραμμα turnin (δείτε οδηγίες στην ιστοσελίδα του μαθήματος), είτε στους βοηθούς του μαθήματος, τη Δευτέρα, 16 Οκτωβρίου 2017, ώρα 15:00-16:00. Ασκήσεις που παραδίδονται μετά τις 16:00 της Δευτέρας, 16/10/2017 θεωρούνται εκπρόθεσμες. Εκπρόθεσμες ασκήσεις γίνονται δεκτές σε ηλεκτρονική μορφή και η αποστολή τους πρέπει να γίνει χρησιμοποιώντας το πρόγραμμα turnin. Εναλλακτικά, εκπρόθεσμες ασκήσεις μπορούν να παραδοθούν στη διδασκουςα.

Άσκηση 1 [20 μονάδες]

- α. Έστω $T_1(n)$ ο συνολικός αριθμός των φορών που εκτελείται η εντολή $x = x+1$ στον παρακάτω αλγόριθμο. Υπολογίστε το $T_1(n)$. Τι τάξης είναι (συμβολισμός $\Theta()$) και γιατί; Δικαιολογήστε την απάντησή σας βρίσκοντας $c_1, c_2 \in \mathbb{R}^+$ και ακέραιους $n_1, n_2 \geq 0$ ώστε να ισχύουν οι ορισμοί. Υποθέστε πως το n είναι άρτιος αριθμός.

```
Procedure HowManyPrimitiveInstructions1(integer n) {                                     [10M]
    int i, j, x = 0;
    for (i = 0; i < n; i = i + 2) {
        for (j = 2i; j < 2n; j=j+1) x = x+1;
    }
}
```

- β. Έστω $T_2(n)$ ο συνολικός αριθμός των φορών που εκτελείται η εντολή $x = x+i$ στον παρακάτω αλγόριθμο. Υπολογίστε το $T_2(n)$. Τι τάξης είναι (συμβολισμός $\Theta()$) και γιατί; Δικαιολογήστε την απάντησή σας βρίσκοντας $c_1, c_2 \in \mathbb{R}^+$ και ακέραιους $n_1, n_2 \geq 0$ ώστε να ισχύουν οι ορισμοί.

```
Procedure HowManyPrimitiveInstructions2(integer n) {                                     [10M]
    for (i = 1; i <=  $\sqrt{n}$ ; i = i+1) {
        x = 1;
        while (x <= 2n) x = x + i;
    }
}
```

Σημείωση: Το $H_n = 1 + 1/2 + 1/3 + \dots + 1/n$ ονομάζεται αρμονικός αριθμός και είναι γνωστό ότι $H_n = O(\ln n)$.

Άσκηση 2 [30 μονάδες]

- α. Έστω $p(n) = a_m n^m + a_{m-1} n^{m-1} + \dots + a_1 n + a_0$ πολώνυμο βαθμού $m > 0$, όπου το m είναι ένας άρτιος αριθμός και για κάθε k , $0 \leq k \leq m$, ισχύει ότι (1) $a_k > 0$ αν το k είναι άρτιος και (2) $a_k < 0$ αν το k είναι περιττός. Να αποδειχθεί ότι $p(n) = \Theta(n^m)$. [10M]

β. Συμπληρώστε στις αγκύλες ([]) την τάξη μεγέθους (O , Ω , Θ) των κάτωθι παραστάσεων: [12M]

(i) $n \log \log n$ = [] ($n \log^2 n$)

(ii) $\sqrt[3]{n \log n}$ = [] ($n^{0.3}$)

(iii) $n \log(\sqrt[6]{n})$ = [] ($n \log n$)

γ. Μελετήστε την ασυμπτωτική τάξη (συμβολισμός Θ) της συνάρτησης: [8M]

$$f(n) = n^3 \log n^3 + n^2 \sqrt{n} - n^2 \log n - 1$$

Άσκηση 3 [25 μονάδες]

Θεωρήστε τον ακόλουθο αλγόριθμο:

```
int ProcessString(char A[], char B[]) { // τα A και B είναι αλφαριθμητικά με το A
                                        // να έχει μεγαλύτερο μήκος από το B
    int n = length(A), m = length(B); // η length() είναι βοηθητική συνάρτηση
                                        // που επιστρέφει το μήκος ενός αλφαριθμητικού
    for (i = 1; i = n-m + 1; i++) {
        j = 1;
        while (j <= m && A[i+j-1] == B[j]) j++;
        if (j == m+1) return i;
    }
    return -1;
}
```

α. Παρουσιάστε γενική περιγραφή της λειτουργίας του παραπάνω αλγορίθμου; Ποιος είναι ο στόχος του (δηλαδή ποιο πρόβλημα λύνει ο αλγόριθμος);

[5M]

β. Ιχνηλατίστε τον αλγόριθμο όταν εκτελείται με είσοδο τα αλφαριθμητικά $A[1,8] = \text{banenana}$ και $B[1,3] = \text{ana}$. [7M]

γ. Μελετήστε τη χρονική πολυπλοκότητα του αλγορίθμου και την τάξη της. Σημειώνεται ότι η χρονική πολυπλοκότητα του αλγορίθμου θα είναι συνάρτηση των n, m . [13M]

Άσκηση 4 [25 μονάδες]

α. Θεωρήστε την αναδρομική σχέση $T(n) = T(n-a) + n + c$. Επιλύστε την αναδρομική σχέση με τη μέθοδο της επαναληπτικής αντικατάστασης και μελετήστε την τάξη της. [10M]

β. Σχεδιάστε το δένδρο αναδρομής που αντιστοιχεί στην αναδρομική σχέση $T(n) = 3T(n/2) + n$. Στη συνέχεια, μελετήστε την τάξη της χρησιμοποιώντας τη μέθοδο της μαθηματικής επαγωγής. [15M]