

HY240: Δομές Δεδομένων

Εαρινό Εξάμηνο – Ακαδημαϊκό Έτος 2014-15

Παναγιώτα Φατούρου

1^η Σειρά Ασκήσεων

Ημερομηνία Παράδοσης: Δευτέρα, 2 Μαρτίου 2015

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, 2 Φεβρουαρίου 2015, ώρα 15:00-16:00. Ασκήσεις που παραδίδονται μετά τις 16:00 της Δευτέρας, 2/3/2015 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 σταλούν στο hy240b@csd.uoc.gr (σε μορφή pdf) μέσω ηλεκτρονικού ταχυδρομείου.

Άσκηση 1 [20 μονάδες]

Έστω $T_1(n)$ το συνολικό πλήθος των φορών που εκτελείται η εντολή $x = x + 3$ από τη ρουτίνα FindMyComplexity1() και $T_2(n)$ το συνολικό πλήθος των φορών που εκτελείται η εντολή $x = x + 3$ από τη ρουτίνα FindMyComplexity2(). Υπολογίστε τα $T_1(n)$ και $T_2(n)$. Μελετήστε την τάξη (συμβολισμός O) των $T_1(n)$ και $T_2(n)$.

<pre> procedure FindMyComplexity1(integer n) for i = 1 to $\sqrt{n}$ do for j = 1 to $\sqrt{n}$ do for k = 1 to $\sqrt{n} - j + 1$ do $x = x + 3$; } }</pre>	<pre> procedure FindMyComplexity2(integer n) for i = 1 to n do if i is odd then { for j = i to n do $x = x + 3$; for j = 1 to i do $x = x + 3$; } }</pre>
--	---

Άσκηση 2 [20 μονάδες]

α. Ποια από τα ακόλουθα είναι αληθή και γιατί;

i. $n^3 - 4n^2 - 5n + 10 = \Theta(n^3)$

ii. $\sum_{i=1}^n (2 * (i + 1)) = O(n^2)$

iii. $\min(700, n^2) = \Theta(1)$

β. Μελετήστε την ασυμπτωτική πολυπλοκότητα (βάσει του συμβολισμού Θ) της συνάρτησης

$$f(n) = 10n^4 \log n^6 + n^4 \sqrt{n} - n \log n - 3$$

Άσκηση 3 [20 μονάδες]

Επιδύστε τις ακόλουθες αναδρομικές σχέσεις με επαναληπτική αντικατάσταση.

α. $T(n) = 4T(n/4) + cn$ [5M]

β. $T(n) = T(n-2) + c$, όπου c είναι μια ακέραια σταθερά. [5M]

Για την αναδρομική σχέση του ερωτήματος α. ζητούνται τα εξής:

i. Σχεδιάστε το δένδρο αναδρομής [5M]

ii. Επαληθεύστε, χρησιμοποιώντας μαθηματική επαγωγή, πως η λύση που βρήκατε με επαναληπτική αντικατάσταση είναι σωστή. [5M]

Άσκηση 4 [10 μονάδες]

Θεωρήστε την ακολουθία αριθμών Fibonacci που ορίζεται ως εξής:

$$F(n) = \begin{cases} 0 & \text{αν } n = 0 \\ 1 & \text{αν } n = 1 \\ F(n-1) + F(n-2) & \text{αν } n > 1 \end{cases}$$

Η αναδρομική υλοποίηση της συνάρτησης αυτής παρουσιάζεται στη συνέχεια:

```
integer Fib(integer n) {  
    int n1, n2;  
    if (n < 2) then return n;  
    else {  
        n1 = Fib(n-1);  
        n2 = Fib(n-2);  
        return (n1+n2);  
    }  
}
```

Ζητείται να ιχνηλατηθεί στο χαρτί (δηλαδή να γίνει trace) η $Fib(n)$ για την περίπτωση όπου $n = 4$. Πρέπει να παρουσιαστούν όλες οι αναδρομικές κλήσεις της Fib καθώς και οι τιμές των μεταβλητών n , $n1$ και $n2$ για κάθε εκτέλεση της Fib . Πρέπει επίσης να παρουσιαστεί ο χώρος στη μνήμη που κατανέμεται για την εκτέλεση των αναδρομικών κλήσεων της Fib .

Άσκηση 5 [30 μονάδες]

Έστω ότι πρέπει να υπολογίσετε την ποσότητα $\sum_{i=1}^n x^i$, $1 \leq i \leq n$. Να βρεθούν δύο τρόποι υπολογισμού της ποσότητας αυτής, οι οποίοι να οδηγούν σε δύο αλγόριθμους με διαφορετική χρονική πολυπλοκότητα. Συγκεκριμένα, ο πρώτος αλγόριθμος θα πρέπει να έχει χρονική πολυπλοκότητα $\Theta(n^2)$ ενώ ο δεύτερος $\Theta(n)$. Παρουσιάστε ψευδοκώδικα για τα δύο αλγορίθμους και **αποδείξτε** πως η πολυπλοκότητά τους είναι η ζητούμενη.