

HY240: Δομές Δεδομένων

Εαρινό Εξάμηνο – Ακαδημαϊκό Έτος 2019-20

Παναγιώτα Φατούρου

1^η Σειρά Ασκήσεων

Ημερομηνία Παράδοσης: Δευτέρα, 24 Φεβρουαρίου 2020

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, 24 Φεβρουαρίου 2020, ώρα 13:00-14:00. Ασκήσεις που παραδίδονται μετά τις 14:00 της Δευτέρας, 24/2/2020 θεωρούνται εκπρόθεσμες. Εκπρόθεσμες ασκήσεις γίνονται δεκτές σε ηλεκτρονική μορφή και η αποστολή τους πρέπει να γίνει χρησιμοποιώντας το πρόγραμμα turnin.

Άσκηση 1 [20 μονάδες]

Έστω $T_1(n)$ το συνολικό πλήθος των φορών που εκτελείται η εντολή $x = x + 1$ από τη ρουτίνα FindMyComplexity1() και $T_2(n)$ το συνολικό πλήθος των φορών που εκτελείται η εντολή $x = x + 1$ 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 n do for j = 1 to $\sqrt{n}$ do for k = 1 to $n - j + 1$ do $x = x + 1$; } }</pre>	<pre> procedure FindMyComplexity2(integer n) for i = 1 to $2n$ do if ($i \% 2 == 1$) then for j = i to $2n$ do $x = x + 1$; }</pre>
---	---

Άσκηση 2 [25 μονάδες]

α. Ποια από τα ακόλουθα είναι αληθή και γιατί;

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

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

iii. $\min(2^{70}, n^3 \log n) = \Theta(1)$

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

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

Άσκηση 3 [25 μονάδες]

Επιλύστε τις ακόλουθες αναδρομικές σχέσεις με επαναληπτική αντικατάσταση.

$$\alpha. T(n) = \begin{cases} 1, & \text{αν } n = 1 \\ 3T\left(\frac{n}{3}\right) + n, & \text{διαφορετικά} \end{cases} \quad [7M]$$

$$\beta. T(n) = \begin{cases} 1, & \text{αν } 0 \leq n < c \\ T(n - c) + n, & \text{διαφορετικά} \end{cases}, \text{ όπου } c \text{ είναι μια ακέραια σταθερά.} \quad [8M]$$

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 [30 μονάδες]

Θεωρήστε τον ακόλουθο αναδρομικό αλγόριθμο, ο οποίος εκτελείται πάνω σ' έναν μη-ταξινομημένο πίνακα ακεραίων $T[s...u]$ με $n = u - s + 1 = 2^k$ ($k > 0$) στοιχεία, όπου $s, u, u > s$, είναι θετικοί ακέραιοι που καθορίζουν το πρώτο και το τελευταίο στοιχείο του πίνακα.

```
int RecursiveF(table T: array of int, int s, int u) {  
    int middle; int val;  
  
    if (u-s == 0) return T[s];  
    middle = ⌊(s+u)/2⌋;  
    for j = s to middle do {  
        if (T[j] > T[j+middle]) then  
            swap(T[j], T[j+middle]);  
    }  
    val = RecursiveF(T, s, middle);  
    return val;  
}
```

α. Τι υπολογίζει η RecursiveF; Παρουσιάστε σύντομη περιγραφή του τρόπου λειτουργίας της. [7M]

β. Ιχνηλατίστε την RecursiveF(T,1,8) για την περίπτωση που $T = [20, 4, 40, 60, 34, 16, 38, 2]$. Παρουσιάστε σύντομη περιγραφή του τρόπου λειτουργίας του αλγορίθμου. [8M]

γ. Διατυπώστε μια αναδρομική σχέση για το χρόνο εκτέλεσης της RecursiveF(). [5M]

δ. Επιλύστε την αναδρομική σχέση που προτείνατε στο ερώτημα γ. και μελετήστε (βάσει των συμβολισμών O και Ω) την τάξη της πολυπλοκότητάς της. [5M]

ε. Ποια είναι η χωρική πολυπλοκότητα του αλγορίθμου (δηλαδή πόση μνήμη απαιτείται για να εκτελεστεί ο αλγόριθμος); [5M]