

HY240: Δομές Δεδομένων

Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2016-17

Παναγιώτα Φατούρου

1^η Σειρά Ασκήσεων

Ημερομηνία Παράδοσης: Δευτέρα, 10 Οκτωβρίου 2016

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, 10 Οκτωβρίου 2016, ώρα 15:00-16:15. Ασκήσεις που παραδίδονται μετά τις 16:15 της Δευτέρας, 10/10/2016 θεωρούνται εκπρόθεσμες. Εκπρόθεσμες ασκήσεις γίνονται δεκτές σε ηλεκτρονική μορφή και η αποστολή τους πρέπει να γίνει χρησιμοποιώντας το πρόγραμμα turnin. **Ασκήσεις μέσω ηλεκτρονικού ταχυδρομείου δεν γίνονται δεκτές.**

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(int n) { for i = 1 to n do for j = n down to i do for k = 1 to n+m do x = x+1; } </pre>	<pre> procedure FindMyComplexity2(int n) { for i = 1 to n² do for j = 1 to 1000 do for (k = 0; k < n²; k+ = n) x = x+1; } </pre>
---	---

Άσκηση 2 [30 μονάδες]

α. Ποιες από τις ακόλουθες προτάσεις είναι αληθείς; Θεωρήστε ότι $f = f(n)$ και $g = g(n)$ είναι συναρτήσεις με πεδίο ορισμού το σύνολο των θετικών αριθμών.

Πρόταση	Αληθής ή Ψευδής?
$n^3 + 4n^2 \log^2 n + 5n \log n + 10n = \Theta(n^3)$	
$n^3 + 4n^2 \log^2 n + 5n \log n + 10n = O(n^3 \log n)$	
$\sqrt{n^3} \log(\sqrt{n^3}) = O(n^2)$	

Αποδείξτε τους ισχυρισμούς σας.

β. Συμπληρώστε τον παρακάτω πίνακα:

Έκφραση	Κυρίαρχος όρος	$O(\dots)$
$2^{10} n^2 \log n + 2^{10000} + 0.0001 n$		

$n^2 2^n + n^{0.8} + 2 n \log(2^{2^n})$		
$500 n + 12 n^{1.5} + 2n^{2.5}$		
$n \log n + 4 n(\log n)^2$		
$5 n \log n + 2 \log n^2$		

γ. Ταξινομήστε τις ακόλουθες συναρτήσεις, σε αύξουσα διάταξη, βάσει της ασυμπτωτικής τους τάξης.

$$f_1(n) = 3^n, \quad f_2(n) = 10 \sqrt{n} \log^2 n, \quad f_3(n) = n^2, \quad f_4(n) = 3^{2^{30000}},$$

$$f_5(n) = n^2 \log n, \quad f_6(n) = n^2 \sqrt{n}, \quad f_7(n) = \prod_{i=1}^n (i+5)$$

Άσκηση 3 [25 Μονάδες]

- α. Δείξτε και με τις δύο μεθόδους (μέθοδος εικασίας και μέθοδος επαναληπτικής αντικατάστασης) που έχετε διδαχθεί στο μάθημα πως η λύση της αναδρομικής σχέσης $T(1) = 2$ και $T(n) = 2T(n/2) + 4n + 2$ είναι $O(n \log n)$.
- β. Να παρουσιαστούν δύο αλγόριθμοι υπολογισμού του $\sum_{i=1}^n x^i$, $1 \leq i \leq n$, οι οποίοι να έχουν χρονική πολυπλοκότητα διαφορετικής τάξης. Ο κάθε αλγόριθμος θα παίρνει ως παράμετρο το x και θα επιστρέφει την τιμή του πολυωνύμου $\sum_{i=1}^n x^i$, $1 \leq i \leq n$, για την δεδομένη τιμή του x . Να θεωρηθεί ότι τα a_i είναι γνωστές σταθερές. Παρουσιάστε ψευδοκώδικα για τους δύο αλγόριθμους και μελετήστε φορμαλιστικά την τάξη της χρονικής τους πολυπλοκότητας (βάσει των συμβολισμών O , Ω και Θ).

Άσκηση 4 [30 μονάδες]

Δίνεται ένας πίνακας $A[1 \dots n]$ για τον οποίο ισχύει ότι υπάρχει ένας ακέραιος $m \in \{1, 2, \dots, n\}$ τέτοιος ώστε:

- $A[i] < A[i+1]$ για κάθε $1 \leq i < m$, και
- $A[i] > A[i+1]$ για κάθε $m \leq i < n$.

Είναι αξιοσημείωτο πως τα στοιχεία του A μέχρι και το m -οστό είναι σε αύξουσα διάταξη ενώ τα στοιχεία του A που έπονται του m -οστού στοιχείου είναι σε φθίνουσα διάταξη. Επομένως, το στοιχείο $A[m]$ είναι το μεγαλύτερο στοιχείο του πίνακα.

- Σχεδιάστε **αναδρομικό αλγόριθμο** που θα υπολογίζει το μέγιστο στοιχείο του A . **Ο αλγόριθμός σας θα πρέπει να έχει χρονική πολυπλοκότητα $O(\log n)$.** [15M]
- Διατυπώστε μια αναδρομική σχέση για το χρόνο εκτέλεσης του αλγορίθμου σας. [5M]
- Επιλύστε την αναδρομική σχέση που προτείνετε στο ερώτημα ii. και μελετήστε την τάξη της πολυπλοκότητας του αλγορίθμου σας. [5M]
- Ποια είναι η χωρική πολυπλοκότητα του αλγορίθμου σας (δηλαδή πόση μνήμη απαιτείται για να εκτελεστεί ο αλγόριθμος); [5M]

Υπόδειξη: Εφόσον για κάθε ακέραιο $1 \leq i < n$, είτε $A[i] < A[i + 1]$ ή $A[i] > A[i + 1]$, ισχύει ότι:

1. Αν $A[i] < A[i + 1]$, τότε το μέγιστο στοιχείο του A βρίσκεται στο κομμάτι $A[i + 1..n]$.
2. Αν όμως $A[i] > A[i + 1]$, τότε το μέγιστο στοιχείο του A βρίσκεται στο κομμάτι $A[1..i]$.

Χρησιμοποιήστε τις παραπάνω παρατηρήσεις για να σχεδιάσετε τον αλγόριθμό σας. Προσοχή στην πολυπλοκότητα που ζητείται.