ΗΥ240: Δομές Δεδομένων
Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2014-15
Παναγιώτα Φατούρου

1η Σειρά Ασκήσεων

Ημερομηνία Παράδοσης: Δευτέρα, 13 Οκτωβρίου 2014
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, 13 Οκτωβρίου 2014, ώρα 15:00-16:00. Ασκήσεις που παραδίδονται μετά τις 16:00 της Δευτέρας, 13/10/2014 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 hy240a@csd.uoc.gr (σε μορφή pdf) μέσω ηλεκτρονικού ταχυδρομείου.

Άσκηση 1 [20 μονάδες]

α. Έστω $T_1(n)$ το συνολικό πλήθος των φορών που εκτελείται η εντολή x = x + 5 από τη ρουτίνα Tricky1 και $T_2(n)$ το συνολικό πλήθος των φορών που εκτελείται η εντολή x = x + 5 από τη ρουτίνα Tricky2. Υπολογίστε τα $T_1(n)$ και $T_2(n)$. Μελετήστε την τάξη (συμβολισμός O) των $T_1(n)$ και $T_2(n)$. [20M]

Procedure Tricky1(integer n) {
int x;
for (i = 0; i <= $3n^2$; i = i + 3) {
for (j = i; j <= $2\sqrt{n}$; j = j + 2)
x = x+5;
}
}
	Procedure Tricky2 (integer n) {
	int x;
	for (j = n; j <= 3n; j++) {
	for (k = 3^{n-1}; k < 3^n; k++) {
	for (m = 0; m < 30; m++)
	x = x+5;
	}
	}
	}

Άσκηση 2 [30 μονάδες]

α. Μελετήστε την ασυμπτωτική πολυπλοκότητα (βάσει του συμβολισμού Θ) των συναρτήσεων που ακολουθούν: [12M]

i. $f(n) = 5\log(n^{10} + 10) + \log 5^{10}$

ii. $f(n) = 3n^5 \log n + 10n^2\sqrt{n} + 4n\log n + 3$

iii. $f(n) = n^5 – 3n^4 \log^2 n - 7n^2 - 10$

β. Ταξινομήστε τις συναρτήσεις κάθε μιας από τις παρακάτω ομάδες συναρτήσεων, σε αύξουσα διάταξη, βάσει της ασυμπτωτικής τους τάξης (συμβολισμός O). [9M]

Group 1: $f_1(n) = (1.1)^n$, $f_2(n) = n^{0.5} \log n$, $f_3(n) = n^2$, $f_4(n) = 100000n$

Group 2: $f_1(n) = 2^{2^{10000000}}$, $f_2(n) = 2^{100000} n$, $f_3(n) = n \log n$, $f_4(n)=n\sqrt{n}$

Group 3: $f_1(n) = n^{\sqrt{n}}$, $f_2(n) = 2^n$, $f_3(n) = \sum_{i=1}^{n}(i+3)$, $f_4(n) = \sum_{i=1}^{n/2}(2i)$,

γ. Θεωρείστε πως μελετάμε συναρτήσεις των οποίων το πεδίο τιμών είναι το R^+. Εξετάστε αν κάθε μια από τις παρακάτω προτάσεις είναι (1) **αληθής (για οποιαδήποτε τέτοια συνάρτηση)**, (2) **ψευδής (για οποιαδήποτε τέτοια συνάρτηση)** ή (3) **για κάποιες συναρτήσεις αληθής και για κάποιες άλλες ψευδής**. Στις δύο πρώτες περιπτώσεις, αποδείξτε την ορθότητα του ισχυρισμού σας. Στην τρίτη περίπτωση, παρουσιάστε ένα παράδειγμα συνάρτησης για την οποία η πρόταση είναι αληθής και ένα παράδειγμα συνάρτησης για την οποία είναι ψευδής. [9M]

i. $f(n) = O((f(n))^3)$

ii. $f(n) + O(f(n)) = \Theta(f(n))$

iii. $f(n) + g(n) = \Theta(\max\{f(n),g(n)\})$

Άσκηση 3 [25 μονάδες]

Επιλύστε τις ακόλουθες αναδρομικές σχέσεις (με επαναληπτική αντικατάσταση ή φτιάχνοντας το δένδρο αναδρομής) και στη συνέχεια για μία από τις αναδρομικές σχέσεις των ερωτημάτων α.-β. καθώς και για την αναδρομική σχέση του ερωτήματος γ. αποδείξτε πως οι λύσεις που βρήκατε είναι σωστές χρησιμοποιώντας μαθηματική επαγωγή: [5M για κάθε ερώτημα + 10 μονάδες για τις επαγωγές]

α. $T(1) = 1$, $T(n) = 3T(n/4) + n$

β. $T(1) = 1$, $T(n) = 2T(n/2) + n^2$

γ. $T(1) = 1$, $T(n) = 2T(n-1) + 1$

Άσκηση 4 [25 μονάδες]

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

i. Σχεδιάστε **αναδρομικό αλγόριθμο** που θα υπολογίζει το μέγιστο στοιχείο του A. **Ο αλγόριθμός σας θα πρέπει να έχει χρονική πολυπλοκότητα O(log n).** [14M]

ii. Διατυπώστε μια αναδρομική σχέση για το χρόνο εκτέλεσης του αλγορίθμου σας. [3M]

iii. Επιλύστε την αναδρομική σχέση που προτείνατε στο ερώτημα ii. και μελετήστε την τάξη της πολυπλοκότητας του αλγορίθμου σας. [3M]

iv. 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.