

HY240: Δομές Δεδομένων

Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2019-20

Παναγιώτα Φατούρου

1^η Σειρά Ασκήσεων

Ημερομηνία Παράδοσης: Δευτέρα, 14 Οκτωβρίου 2019

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, 14 Οκτωβρίου 2019, ώρα 15:00-16:00. Ασκήσεις που παραδίδονται μετά τις 16:00 της Δευτέρας, 14/10/2019 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 [15 μονάδες]

Έστω $T_1(n)$ το συνολικό πλήθος των φορών που εκτελείται η εντολή $x = x + 3$ από τη ρουτίνα Tricky1 και $T_2(n)$ το συνολικό πλήθος των φορών που εκτελείται η εντολή $x = x + 3$ από τη ρουτίνα Tricky2. Υπολογίστε τα $T_1(n)$ και $T_2(n)$. Μελετήστε την τάξη (συμβολισμός Θ) των $T_1(n)$ και $T_2(n)$. [7 M + 8 M]

<pre> Procedure Tricky1(integer n) { int x; for (i = 0; i <= 5n²; i = i + 1) { for (j = i; j <= 3√n; j = j + 2) x = x+3; } } </pre>	<pre> Procedure Tricky2 (integer n) { int x; for (j = n; j <= 2n; j = j+1) { for (k = 5ⁿ; k < 5ⁿ⁺¹; k = k+1) { for (m = k; m < k+10; m = m+1) x = x+3; } } } </pre>
--	--

Άσκηση 2 [30 μονάδες]

α. Θεωρήστε δύο συναρτήσεις $f(n)$ και $g(n)$. Χαρακτηρίστε, ως αληθή ή ψευδή, κάθε μια από τις παρακάτω εκφράσεις που αφορούν τον συμβολισμό $O()$. Αποδείξτε τους ισχυρισμούς σας. Για τις εκφράσεις που δεν είναι αληθείς, προτείνετε μια εναλλακτική έκφραση που να είναι αληθής και αποδείξτε πως η έκφραση αυτή είναι πράγματι αληθής. Επιπρόσθετα, αν μια έκφραση είναι αληθής για κάποιες συναρτήσεις και ψευδής για άλλες, παρουσιάστε ένα παράδειγμα συνάρτησης για την οποία η έκφραση είναι αληθής και ένα παράδειγμα συνάρτησης για την οποία είναι ψευδής. [18M]

- i. Αθροιστικός Κανόνας: $O(f + g) = O(f) + O(g)$
- ii. Πολλαπλασιαστικός Κανόνας: $O(f \cdot g) = O(f) \cdot O(g)$

iii. Μεταβατικός Κανόνας: if $g = O(f)$ and $h = O(f)$ then $g = O(h)$

iv. $5n + 8n^2 \log n + 100n^3 = O(n^4)$

v. $5n + 8n^2 \log(n^2) + 100n^3 = O(n^2 \log n)$

β. Δύο πακέτα λογισμικού A και B με χρονική πολυπλοκότητα $O(n \log n)$ και $O(n)$, αντίστοιχα, χρειάζονται ακριβώς $T_A(n) = c_A n \log_{10} n$ milliseconds and $T_B(n) = c_B n$ milliseconds για να επεξεργαστούν n στοιχεία (δηλαδή για να επεξεργαστούν είσοδο μεγέθους n). Κατά τη διάρκεια ενός πειράματος, διαπιστώθηκε ότι για την επεξεργασία 10^4 στοιχείων, το πακέτο A απαιτεί 100 milliseconds, ενώ το πακέτο B απαιτεί 500 milliseconds. Παρουσιάστε ακριβείς συνθήκες υπό τις οποίες το κάθε πακέτο έχει καλύτερο χρόνο εκτέλεσης από το άλλο. Ποιο πακέτο είναι η καλύτερη επιλογή για την επεξεργασία $n = 10^9$ στοιχείων και γιατί; [4M]

γ. Αποδείξτε ότι η συνάρτηση $T(n) = a_0 + a_1 n + a_2 n^2 + a_3 n^3$ είναι στο $\Theta(n^3)$ χρησιμοποιώντας τον ορισμό του συμβολισμού Θ . Θεωρήστε ότι τα a_0, a_1, a_2 είναι σταθερές στο $\mathbb{Z}$ (δηλαδή μπορεί να είναι είτε θετικοί ή αρνητικοί ακέραιοι αριθμοί), ενώ το $a_3 > 0$, είναι θετικός ακέραιος. [8M]

Άσκηση 3 [20 μονάδες]

Επλύστε 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.-γ αποδείξτε πως οι λύσεις που βρήκατε είναι σωστές χρησιμοποιώντας μαθηματική επαγωγή:

[5M για κάθε ερώτημα + 5 μονάδες για την επαγωγή]

α. $T(1) = 1, T(n) = 2T(n/4) + 1$

β. $T(1) = 1, T(n) = 4T(n/4) + n$

γ. $T(1) = 1, T(n) = 2T(n-5) + n$

Υπόδειξη: Για κάθε πραγματικό αριθμό $q \neq 1$ ισχύει
$$\sum_{k=1}^n kq^k = \frac{nq^{n+1}}{q-1} - \frac{q^{n+1} - q}{(q-1)^2}.$$

Άσκηση 4 [35 μονάδες]

Δίνεται ένας μη-ταξινομημένος πίνακας $A[1 \dots n]$. Ζητείται να σχεδιάσετε αναδρομικό αλγόριθμο που θα βρίσκει το k -οστό μικρότερο στοιχείο στον πίνακα A. Ο αλγόριθμος σας θα πρέπει να λειτουργεί ως εξής:

Αρχικά ο αλγόριθμος επιλέγει, ένα αυθαίρετο στοιχείο, το οποίο ονομάζεται στοιχείο οδηγός (pivot). Στη συνέχεια, ο πίνακας χωρίζεται σε δύο υπο-πίνακες, με τον πρώτο να περιέχει όλα τα στοιχεία που είναι μικρότερα από ή ίσα με το στοιχείο οδηγό και τον δεύτερο να περιέχει εκείνα που είναι μεγαλύτερα απ' αυτό. Έστω m το πλήθος των στοιχείων στον πρώτο υπο-πίνακα. Αν $m \geq k$, τότε γνωρίζουμε ότι το k -οστό μικρότερο στοιχείο από ολόκληρο τον πίνακα είναι επίσης το k -οστό μικρότερο στοιχείο από τον πρώτο υπο-πίνακα. Κατά συνέπεια, μπορούμε να λύσουμε το πρόβλημα, δουλεύοντας στον πρώτο υπο-πίνακα. Διαφορετικά, αν $m < k$, τότε το k -οστό μικρότερο στοιχείο από ολόκληρο τον πίνακα είναι το $(k-m)$ -οστό μικρότερο στοιχείο από τον δεύτερο υπο-πίνακα.

- i. Σχεδιάστε **αναδρομικό αλγόριθμο** που θα υπολογίζει το k -οστό μικρότερο στοιχείο στον πίνακα A υλοποιώντας την παραπάνω τεχνική. [10M]
- ii. Θεωρήστε ότι είστε τυχεροί και ότι το στοιχείο οδηγός είναι πάντα το διάμεσο: αν ο πίνακας έχει n στοιχεία, το διάμεσο είναι το $(n/2)$ -οστό μικρότερο στοιχείο του πίνακα. Διατυπώστε μια αναδρομική σχέση για το χρόνο εκτέλεσης του αλγορίθμου σας σ' αυτήν την περίπτωση. [5M]
- iii. Επιλύστε την αναδρομική σχέση που προτείνετε στο ερώτημα ii. και μελετήστε την τάξη της πολυπλοκότητας του αλγορίθμου σας. [5M]
- iv. Ποια είναι η χειρότερη περίπτωση για τον παραπάνω αλγόριθμο; Διατυπώστε μια αναδρομική σχέση για το χρόνο εκτέλεσης του αλγορίθμου σας στη χειρότερη περίπτωση. [5M]
- v. Επιλύστε την αναδρομική σχέση που προτείνετε στο ερώτημα iv. και μελετήστε την τάξη της πολυπλοκότητας του αλγορίθμου σας. [5M]
- vi. Ποια είναι η χωρική πολυπλοκότητα του αλγορίθμου σας (δηλαδή πόση μνήμη απαιτείται για να εκτελεστεί ο αλγόριθμος) σε κάθε μια από τις παραπάνω δύο περιπτώσεις; [5M]