ΗΥ240: Δομές Δεδομένων
Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2018-19
Παναγιώτα Φατούρου

1η Σειρά Ασκήσεων

Ημερομηνία Παράδοσης: Δευτέρα, 15 Οκτωβρίου 2018

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, 15 Οκτωβρίου 2018, ώρα 15:00-16:00. Ασκήσεις που παραδίδονται μετά τις 16:00 της Δευτέρας, 15/10/2018 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 διδάσκουσα.

Άσκηση 1 [20 μονάδες]

α. Έστω $T_1(n)$ ο συνολικός αριθμός των φορών που εκτελείται η εντολή x= x+1 στον παρακάτω αλγόριθμο. Υπολογίστε το $T_1(n)$. Τι τάξης είναι (συμβολισμός Θ()) και γιατί; Δικαιολογήστε την απάντησή σας βρίσκοντας $c_1, c_2 \in R^+$ και ακέραιους $n_1, n_2 \geq 0$ ώστε να ισχύουν οι ορισμοί. Υποθέστε πως το n είναι άρτιος αριθμός.

Procedure **Puzzle1**(integer n) { [10M]

```
    int i, j, x = 0;
    for (i = 0; i < √n ; i = i + 1)  {
            for (j = n; j >= i; j=j-1) x = x+1;
    }
}
```

β. Έστω $T_2(n)$ ο συνολικός αριθμός των φορών που εκτελείται η εντολή x = x+1 στον παρακάτω αλγόριθμο. Υπολογίστε το $T_2(n)$. Τι τάξης είναι (συμβολισμός Θ()) και γιατί; Δικαιολογήστε την απάντησή σας βρίσκοντας $c_1, c_2 \in R^+$ και ακέραιους $n_1, n_2 \geq 0$ ώστε να ισχύουν οι ορισμοί.

Procedure **Puzzle2**(integer n) { [10M]

```
    int i, j, x = 0;
    for (i = 0; i < n^2; i = i + 1)  {
            for (j = n√n ; j >= i; j=j-1)
                    for (k = 0; k < 2^n; k++) x = x+1;
    }
}
```

Άσκηση 2 [30 μονάδες]

α. (i) Αποδείξτε, χρησιμοποιώντας τη μέθοδο της μαθηματικής επαγωγής, ότι: [5M]

$$\sum_{j=0}^{n} ar^{j} = \frac{ar^{n+1} - a}{r-1}$$

(ii) Χρησιμοποιήστε τη μέθοδο της μαθηματικής επαγωγής για να αποδείξετε ότι κάθε ακέραιος αριθμός n > 1 μπορεί να εκφραστεί ως γινόμενο πρώτων αριθμών. [5M]

Υπόδειξη: Διακρίνετε περιπτώσεις για το αν ο αριθμός που εξετάζετε στο επαγωγικό βήμα είναι πρώτος ή όχι.

β. (i) Αποδείξτε ότι $T(n) = (n^3+2n)/(2n+1) \in O(n^2)$. [4M]

(ii) Αποδείξτε ότι $f(n) = \Theta(g(n))$ αν και μόνο αν $f(n) = O(g(n))$ και $g(n) = O(f(n))$. [4M]

(iii) Αποδείξτε ότι για κάθε ζεύγος πραγματικών αριθμών a > 1 και b > 1, αν $f(n) = O(\log_b n)$, τότε $f(n) = O(\log_a n)$. [4M]

γ. Μελετήστε την ασυμπτωτική τάξη (συμβολισμός Θ) των ακόλουθων συναρτήσεων:

(i) $T(n) = (17 \log n +19)(n^3+2) - (n^3+ n^2 \log n)(\log n +1)$ [4M]

(ii) $T(n) = 2(n \log n + 1)^2 - (\log n +1)(n^2+1)$ [4M]

Άσκηση 3 [20 μονάδες]

Παρουσιάστε έναν αλγόριθμο, που θα ονομάζεται Partition-Even-Odd(int A[]) και θα διαχωρίζει έναν πίνακα A έτσι ώστε όλα τα στοιχεία του πίνακα που είναι άρτια να βρίσκονται στις πρώτες θέσεις του πίνακα, ενώ εκείνα που είναι περιττά να βρίσκονται στις τελευταίες θέσεις του πίνακα (δηλαδή να εμφανίζονται στον πίνακα σε θέσεις μετά από όλα τα άρτια στοιχεία). Για παράδειγμα, θεωρήστε τον πίνακα A = [7, 17, 74, 21, 7, 9, 26, 10]. Μετά την εκτέλεση του αλγορίθμου σας, θα πρέπει ο πίνακας να έχει την εξής μορφή: A = [74, 26, 10, 21, 7, 9, 17, 7]. **Η χωρική επιβάρυνση του αλγορίθμου σας θα πρέπει να είναι O(1)** (το οποίο σημαίνει ότι ο αλγόριθμός σας επιτρέπεται να χρησιμοποιεί μόνο έναν σταθερό αριθμό από βοηθητικές μεταβλητές επιπρόσθετα του A και επομένως δεν επιτρέπεται να χρησιμοποιεί βοηθητικούς πίνακες). Επίσης, **ο αλγόριθμός σας, θα πρέπει να έχει χρονική πολυπλοκότητα O(n)**, όπου n είναι το πλήθος των στοιχείων του πίνακα.

(α) Παρουσιάστε ψευδοκώδικα για τον αλγόριθμο Partition-Even-Odd().

(β) Μελετήστε την χρονική πολυπλοκότητα του αλγορίθμου σας.

Άσκηση 4 [30 μονάδες]

α. Θεωρήστε τον ακόλουθο αλγόριθμο: [20M]

```
procedure Mystery(integer A[], integer i, integer j, integer x) {
    if (A[i] == x) then y = i;
    else if (i == j) then y = -1;
    else y = Mystery(A, i+1, j, x);
    return y;
}
```

i. Ιχνηλατίστε την Mystery(A, 0, 7, 26) για την περίπτωση που A = [7, 17, 74, 21, 7, 9, 26, 10] και x = 26.

ii. Εξηγήστε ποιο πρόβλημα επιλύει η Mystery(). Παρουσιάστε σύντομη περιγραφή του τρόπου λειτουργίας της.

iii. Διατυπώστε μια αναδρομική σχέση για το χρόνο εκτέλεσης της Mystery().

iv. Επιλύστε την αναδρομική σχέση που προτείνατε στο ερώτημα iii. και μελετήστε την τάξη της χρονικής πολυπλοκότητας της Mystery().

v. Ποια είναι η χωρική πολυπλοκότητα της Mystery() (δηλαδή πόση μνήμη απαιτείται για να εκτελεστεί);

β. Θεωρήστε την αναδρομική σχέση $T(n) = 4T(n/4) + 1$. Θεωρήστε ότι $T(1) = 1$. [10M]

i. Σχεδιάστε το δένδρο αναδρομής για την T(n).

ii. Βρείτε κλειστό τύπο για την T(n) χρησιμοποιώντας τη μέθοδο της επαναληπτικής αντικατάστασης.

iii. Μελετήστε την ασυμπτωτική συμπεριφορά της T(n) χρησιμοποιώντας τη μέθοδο της εικασίας και της μαθηματικής επαγωγής για την επαλήθευση της εικασίας σας.