



HY240 — ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

1η Σειρά Ασκήσεων

Τρόπος Παράδοσης

Οι ασκήσεις μπορούν να παραδοθούν στους βοηθούς του μαθήματος, τη **Δευτέρα, 16 Οκτωβρίου 2023, ώρα 11:00-11:59 π.μ.** στο γραφείο τους (**B.208 / B.210**). Ασκήσεις που παραδίδονται μετά τις 12:00 μ.μ. της Δευτέρας, 16/10/2023 θεωρούνται εκπρόθεσμες. **Εκπρόθεσμες ασκήσεις γίνονται δεκτές μόνο σε ηλεκτρονική μορφή** και η αποστολή τους πρέπει να γίνει χρησιμοποιώντας το πρόγραμμα turnin. Για περισσότερες πληροφορίες ανατρέξτε στην ιστοσελίδα του μαθήματος.

ΕΞΑΜΗΝΟ:

Εαρινό (2023-24)

ΠΑΝΕΠΙΣΤΗΜΙΟ:

Πανεπιστήμιο Κρήτης

ΤΜΗΜΑ:

Επιστήμης Υπολογιστών

ΚΑΘΗΓΗΤΗΣ/ΤΡΙΑ:

Παναγιώτα Φατούρου

ΥΠΕΥΘΥΝΟΙ ΒΟΗΘΟΙ:

Ιορδάνης Σαπίδης

ΤΕΛΕΥΤΑΙΑ ΤΡΟΠΟΠΟΙΗΣΗ ΕΚΦΩΝΗΣΗΣ:

10 / 10 / 2023

Άσκηση 1

[20 μονάδες]

Έστω $T_1(n)$ το συνολικό πλήθος των φορών που εκτελείται η εντολή $x = x + 3$ από τη ρουτίνα `Puzzle1()` και $T_2(n)$ το συνολικό πλήθος των φορών που εκτελείται η εντολή $x = x + 3$ από τη ρουτίνα `Puzzle2()`. Υπολογίστε τα $T_1(n)$ και $T_2(n)$. Μελετήστε την τάξη (συμβολισμός O) των $T_1(n)$ και $T_2(n)$.

```
procedure Puzzle1(integer n) {  
    for i = 1 to sqrt(n) do  
        for j = i to n do  
            for k = 1 to 2n do  
                x = x+3;  
            }  
        }  
    }  
}
```

```
procedure Puzzle2(integer n)  
    for i = 0 to 2n do  
        if i is even then {  
            for j = i to n do x = x+3;  
        }  
    }  
}
```

Άσκηση 2

[25 μονάδες]

a. Ποια από τα ακόλουθα είναι αληθή και γιατί;

• $n^2 - 2n - 7 = \Theta(n^2)$ [5M]

• $n \times \sum_{i=1}^n (3 \times (i + 1)) = O(n^4)$ [5M]

• $n^2 \log n^2 - n \log n - n = \Theta(n^2 \log n)$ [5M]

b. Μελετήστε την ασυμπτωτική πολυπλοκότητα (βάσει του συμβολισμού Θ) της παρακάτω συνάρτησης:

$$f(n) = 10n^3 \log n^6 + n^3 \sqrt{n} - n \log n - 5 \quad [10M]$$

Άσκηση 3

[20 μονάδες]

Επιλύστε τις ακόλουθες αναδρομικές σχέσεις με επαναληπτική αντικατάσταση.

a. $T(n) := \begin{cases} 1, & \text{για } n \in \{0, 1\} \\ 3T(\lfloor \frac{n}{3} \rfloor) + 3n, & \text{διαφορετικά} \end{cases}$ [5M]

b. $T(n) := \begin{cases} 1, & \text{για } n \leq 0 \\ T(n - c) + c \cdot n, & \text{διαφορετικά, με το } c \text{ να είναι μια ακέραια σταθερά} \end{cases}$ [5M]

Για την αναδρομική σχέση του ερωτήματος a. ζητούνται τα εξής:

c. Σχεδιάστε το δένδρο αναδρομής [5M]

d. Επαληθεύστε, χρησιμοποιώντας μαθηματική επαγωγή, πως η λύση που βρήκατε με επαναληπτική αντικατάσταση είναι σωστή. [5M]

Άσκηση 4

[15 μονάδες]

Δίνεται η παρακάτω συνάρτηση, η οποία εκτελεί Αιγυπτιακό πολλαπλασιασμό:

$$m(x, y) := \begin{cases} 0, & \text{αν } y = 0 \\ m(x + x, \frac{y}{2}), & \text{αν } y \text{ άρτιος και } y \neq 0 \\ x + m(x, y - 1), & \text{αν } y \text{ περιττός και } y \neq 0 \end{cases}$$

- a. Παρουσιάστε ψευδοκώδικα για μια αναδρομική συνάρτηση [5M]

```
int EgyptianMultiplication(int x, int y)
```

η οποία θα πραγματοποιεί αιγυπτιακό πολλαπλασιασμό σύμφωνα με τον παραπάνω αναδρομικό τύπο.

- b. Ιχνηλατίστε την εκτέλεση της `EgyptianMultiplication(5,20)` (δηλαδή παρουσιάστε αναλυτικά την εκτέλεσή της, καθώς και τα activation records στη μνήμη, όταν καλείτε με $x = 5$ και $y = 20$). [10M]

Άσκηση 5

[20 μονάδες]

Η `InsertionSort()` μπορεί να εκφραστεί ως αναδρομική διαδικασία με τον ακόλουθο τρόπο. Για να ταξινομήσουμε τον πίνακα $A[1..n]$, ταξινομούμε αναδρομικά τον υποπίνακα $A[1..n-1]$ και στη συνέχεια ενθέτουμε το στοιχείο $A[n]$ στον ταξινομημένο υποπίνακα $A[1..n-1]$.

- Παρουσιάστε ψευδοκώδικα για την αναδρομική αυτή έκδοση της `InsertionSort()`.
- Διατυπώστε μια αναδρομική σχέση για το χρόνο εκτέλεσης της αναδρομικής αυτής έκδοσης της `InsertionSort()`.
- Επιλύστε την αναδρομική σχέση που προτείνετε στο ερώτημα b. και μελετήστε (βάσει των συμβολισμών O και Ω) την τάξη της πολυπλοκότητας της αναδρομικής έκδοσης της `InsertionSort()`.
- Ποια είναι η χωρική πολυπλοκότητα του αλγορίθμου (δηλαδή πόση μνήμη απαιτείται για να εκτελεστεί ο αλγόριθμος);