

HY240: Δομές Δεδομένων

Εαρινό Εξάμηνο – Ακαδημαϊκό Έτος 2014-15

Παναγιώτα Φατούρου

1^η Σειρά Ασκήσεων

Ημερομηνία Παράδοσης: Δευτέρα, 12 Οκτωβρίου 2015

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, 12 Οκτωβρίου 2015, ώρα 15:00-16:15. Ασκήσεις που παραδίδονται μετά τις 16:15 της Δευτέρας, 12/10/2015 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 hy240b@csd.uoc.gr (σε μορφή pdf) μέσω ηλεκτρονικού ταχυδρομείου.

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 $T_2(n)$. Για να βρείτε την πολυπλοκότητα της FindMyComplexity2(), θεωρήστε ότι το n είναι ίσο με 2^k , όπου k είναι κάποιος ακέραιος (δηλαδή το n είναι δύναμη του 2).

<pre> procedure FindMyComplexity1(integer n) for i = 1 to n do for j = 1 to $\sqrt{n}$ do for k = 1 to n - j + 1 do x = x+1; } }</pre>	<pre> procedure FindMyComplexity2(integer n) for j = 1 to n do { h=n; while(h > 0) do { for i = 1 to $\sqrt{n}$ do x = x+1; h = h/2; } }</pre>
---	--

Άσκηση 2 [30 μονάδες]

α. Ποιες από τις ακόλουθες προτάσεις είναι αληθείς; Θεωρήστε ότι $f \equiv f(n)$ και $g \equiv g(n)$ είναι συναρτήσεις που παίρνουν θετικές τιμές.

Πρόταση	Αληθής ή Ψευδής?
$O(f+g) = O(f) + O(g)$	
$O(f \cdot g) = O(f) \cdot O(g)$	
$10n + 20n^2 + 10n^3 = O(n^4)$	
$5n + 8n^2 + 20n^3 = O(n^2 \log n)$	

β. Αποδείξτε τους ισχυρισμούς σας.

γ. Συμπληρώστε τον παρακάτω πίνακα:

Έκφραση	Κυρίαρχος όρος	O(...)
$5 + 0.03 n^3 + 0.34 n$		
$50n + 10 n^{1.5} + 3 n \log_{10} n$		
$0.4n + 3n^{1.5} + 2.5n^{1.75}$		
$n \log n + n(\log n)^2$		
$n \log n + \log n^2$		
$0.00004 \log n + 0.01 \log \log n$		
$10n^5 \log n + 20n^2 \sqrt{n} + n \log n + 1$		

Άσκηση 3 [25 μονάδες]

1. Εξηγήστε πως λειτουργούν οι παρακάτω αναδρομικοί αλγόριθμοι:

```

a. int mystery1(int n) {
    if(n==0) return 0;
    y = mystery1(n-1);
    return (y + n);
}

b. void mystery2(char[] a, int length) {
    if (length == 1) {
        print(a);
        return;
    }
    for (int i = 0; i < length; i++) {
        swap(a, i, length-1);
        mystery2(a, length-1);
        swap(a, i, length-1);
    }
}

void swap(char[] a, int i, int j) {
    char c;
    c = a[i];
    a[i] = a[j];
    a[j] = c;
}

```

2. Ιχνηλατίστε την εκτέλεση της `mystery1(5)`. Ζωγραφίστε τη μνήμη και δείξτε τις εγγραφές των αναδρομικών κλήσεων (activation records) που δεσμεύονται και αποδεσμεύονται κατά τη διάρκεια εκτέλεσης της `mystery1()` στη μνήμη. [5%]

3. Ιχνηλατήστε την εκτέλεση της `mystery2(«abc», 3)`. Ζωγραφίστε τη μνήμη και δείξτε τις εγγραφές των αναδρομικών κλήσεων (activation records) που δεσμεύονται και αποδεσμεύονται κατά τη διάρκεια εκτέλεσης της `mystery2()` στη μνήμη. [5%]
4. Παρουσιάστε αναδρομική σχέση που να περιγράφει την χρονική πολυπλοκότητα $T(n)$ της `mystery1()`. Τι τάξης είναι η πολυπλοκότητα της `mystery1()`; Αποδείξτε τον ισχυρισμό σας. [10%]
5. Παρουσιάστε αναδρομική σχέση που να περιγράφει τη χρονική πολυπλοκότητα $T(n)$ της `mystery1()`. Επιλύστε την αναδρομική σχέση με τη μέθοδο της επαναληπτικής αντικατάστασης. [5%]

Άσκηση 4 [25 μονάδες]

Θεωρήστε έναν πίνακα X με n θέσεις οι οποίες περιέχουν πραγματικούς αριθμούς. Ζητούνται αλγόριθμοι που θα υπολογίζουν ένα δεύτερο πίνακα A των n θέσεων για τον οποίο θα ισχύει ότι, για κάθε j , $0 \leq j < n$, το στοιχείο $A[j]$ αποθηκεύει το μέσο όρο $(X[0] + X[1] + \dots + X[j]) / (j+1)$ των $(j+1)$ πρώτων στοιχείων του X . Επομένως, θα ισχύει ότι:

$$A[0] = X[0]$$

$$A[1] = (X[0] + X[1]) / 2$$

$$A[2] = (X[0] + X[1] + X[2]) / 3$$

...

$$A[n-1] = (X[0] + X[1] + \dots + X[n-1]) / n$$

Η ακολουθία των j πρώτων τιμών του X ονομάζεται το j -πρόθεμα του X . Το άθροισμα $X[0] + \dots + X[j]$ ονομάζεται το άθροισμα των στοιχείων του j -προθέματος του X και η τιμή του $A[j]$ ονομάζεται μέσος όρος του j -προθέματος.

1. Παρουσιάστε **δύο αλγόριθμους με διαφορετική χρονική πολυπλοκότητα** που να υλοποιούν τη συνάρτηση `computePrefixAverage(array X)` η οποία θα δημιουργεί τον πίνακα A δοθέντος του πίνακα X . Ο ένας από τους δύο αλγορίθμους που θα παρουσιάσετε θα πρέπει να είναι **ο πιο αποδοτικός** που μπορείτε να σκεφτείτε για την επίλυση αυτού του προβλήματος.
2. Αναλύστε την πολυπλοκότητα των αλγορίθμων που προτείνετε.