HY240: Δομές Δεδομένων
Εαρινό Εξάμηνο – Ακαδημαϊκό Έτος 2015-16
Παναγιώτα Φατούρου

1η Σειρά Ασκήσεων

Ημερομηνία Παράδοσης: Δευτέρα, 29 Φεβρουαρίου 2016
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, 29 Φεβρουαρίου 2016, ώρα 11:00-12:00. Ασκήσεις που παραδίδονται μετά τις 12:00 της Δευτέρας, 29/2/2015 θεωρούνται εκπρόθεσμες. Εκπρόθεσμες ασκήσεις γίνονται δεκτές σε ηλεκτρονική μορφή και η αποστολή τους πρέπει να γίνει χρησιμοποιώντας το πρόγραμμα turnin. **Λύσεις ασκήσεων που αποστέλλονται μέσω ηλεκτρονικού ταχυδρομείου δεν θα γίνονται δεκτές.**

Άσκηση 1 [20 μονάδες]

Έστω $T_1(n)$ το συνολικό πλήθος των φορών που εκτελείται η εντολή x = x + 1 από τη ρουτίνα FindMyCompexity1() 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 $T_2(n)$. [10M + 10M]

```
Procedure FindMyCompexity1(integer n) {
    int i, j, x = 0;
    for (i=0; i <= 4n; i = i + 4)  {
            for (j = i; j <= 2n; j++)
                x = x+1;
    }
}
```

```
procedure FindMyComplexity2(integer n)
    int x;
    for (j = 1; j ≤ n; j++) {
        for (k = 2^(n-1); k < 2^n; k++) {
            for (m = 100; m < 300; m++)
                x = x+1;
        }
    }
}
```

Άσκηση 2 [40 μονάδες]

α. Μελετήστε αυστηρά ασυμπτωτικά φράγματα για τις παρακάτω συναρτήσεις αφού πρώτα τις απλοποιήσετε όσο είναι εφικτό. [3*5M]

Συνάρτηση
$15n \log^2(n^{10}) + 5n\sqrt{n} \log n$
$10n^2 2^{\log n} + 5n^{5/2}\sqrt{n} \log(n^4)$
$\log(5^n) \log n + 4n + 3 \log(n^3)$

β. Για κάθε μια από τις αναδρομικές σχέσεις που παρουσιάζονται στον παρακάτω πίνακα, παρουσιάστε αυστηρό ασυμπτωτικό φράγμα για την T(n), χρησιμοποιώντας τη μέθοδο της επαναληπτικής αντικατάστασης. Για κάθε αναδρομική σχέση T, θεωρήστε ότι T(1) = 1 και c είναι μια σταθερά. [2*5M]

Αναδρομική Σχέση
$T(n) = T(n/2) + cn$
$T(n) = 4T(n/4) + c$

Για τη δεύτερη αναδρομική σχέση ζητούνται τα εξής:

i. Σχεδιάστε το δένδρο αναδρομής. [5M]

ii. Επαληθεύστε, χρησιμοποιώντας μαθηματική επαγωγή, πως η λύση που βρήκατε με επαναληπτική αντικατάσταση είναι σωστή. [10M]

Άσκηση 3 [20 μονάδες]

Θεωρήστε τον αλγόριθμο Peculiar που παρουσιάζεται στη συνέχεια.

```
Algorithm Peculiar(int A[1..n]) {
    int key, i, j;
    for (i = 2; i ≤ n; i++) {
          key = A[i];
          p = BinarySearch(A[1..i], key);
          for (j = i-1; j ≥ p; j--) A[j+1] = A[j];
          A[p] = key;
    }
    return A;
}

int BinarySearch (int A[1..n], int K)
   int left, right, middle;

    left = 1;
    right = n;
    repeat forever {
          if (right < left) then return left;
          else {
               middle = ⌊(left+right)/2⌋;
               if (K == A[middle]->Key) then return middle;
               else if (K < A[middle]->Key) then right = middle-1;
               else left = middle+1;
```

α. Παρουσιάστε σύντομη περιγραφή του ψευδοκώδικα και εξηγήστε τι επιτυγχάνει. [3M]

β. Υπολογίστε την τάξη του πλήθους των συγκρίσεων που θα πραγματοποιηθούν, στη χειρότερη περίπτωση από όλα τα στιγμιότυπα της BinarySearch() που καλεί ο αλγόριθμος. [7M]

γ. Πόσες φορές θα εκτελεστεί η εντολή A[j+1] = A[j] στη χειρότερη περίπτωση; [5M]

δ. Μελετήστε τη χρονική πολυπλοκότητα του αλγορίθμου. [5M]

Σημείωση: Δίνεται ότι $\log (n!) \leq \log n^n \leq n \log n$.

Άσκηση 4 [20 μονάδες]

Θεωρήστε τον ακόλουθο αναδρομικό αλγόριθμο, ο οποίος εκτελεί κάποιο υπολογισμό πάνω σε ένα **μη-ταξινομημένο** πίνακα ακεραίων A[1 .. n] με $n = 2^k$, όπου k είναι κάποιος ακέραιος.

```
int WhatDoICalculate(int A[], int n)  {

   int middle, val, j;

   if (n == 1) return A[1];
   middle = n/2;
   for j = 1 to middle do  {
        if (A[j] > A[j+middle]) then
                swap(&(A[j]),&(A[j+middle]));
   }
   val = WhatDoICalculate(A, middle);
   return val;
}
```

α. Τι υπολογίζει ο αλγόριθμος WhatDoICalculate; Παρουσιάστε σύντομη περιγραφή του τρόπου λειτουργίας του αλγορίθμου. [10M]

β. Παρουσιάστε αναδρομική σχέση που να περιγράφει τη χρονική πολυπλοκότητα της WhatDoICalculate (). Τι τάξης είναι η πολυπλοκότητα της WhatDoICalculate () και γιατί; Δίνεται πως η swap() εκτελεί c στοιχειώδεις εντολές, όπου c είναι μια σταθερά, και ανταλλάσσει την τιμή της μεταβλητής στην οποία δείχνει η πρώτη παράμετρός της με εκείνη της μεταβλητής στην οποία δείχνει η δεύτερη. [10M]