CS240: Data Structures
Winter Semester – Academic Year 2017-18
Panagiota Fatourou

1st Set of Theoretical Exercises

Submission deadline: Monday, 16 October 2017
How to submit: The assignments must be submitted either in electronic format using the turnin program (see instructions on the website of the course), or to the teaching assistants of the course, on Monday, 16 October 2017, time 15:00-16:00. Assignments submitted after 16:00 of Monday, 16/10/2017 are considered out of date. Out of date assignments are accepted only in electronic format using the turnin program.

Exercise 1 [20 points]

a. Let $T_1(n)$ be the total number of times that the x= x+1 instruction is executed in the following algorithm. Find a closed type for $T_1(n)$. What is the order of $T_1(n)$ (in $\Theta()$ notation)? Justify your answer, i.e. find $c_1, c_2 \in R^+$ and an integer $n_0 \geq 0$ so that the conditions of the $\Theta()$ definition are satisfied. Assume that n is an even number.

[10P]

```
Procedure HowManyPrimitiveInstructions1(integer n) {
    int i, j, x = 0;
    for (i = 0; i < n; i = i + 2)  {
            for (j = 2i; j < 2n; j=j+1) x = x+1;
    }
}
```

b. Let $T_2(n)$ be the total number of times that the x= x+i instruction is executed in the following algorithm. Find a closed type for $T_2(n)$. What is the order of $T_2(n)$ (in $\Theta()$ notation)? Justify your answer, i.e. find $c_1, c_2 \in R^+$ and an integer $n_0 \geq 0$ so that the conditions of the $\Theta()$ definition are satisfied.

[10P]

```
Procedure HowManyPrimitiveInstructions2(integer n) {
    for (i = 1; i <= √n ; i = i+1) {
            x = 1;
            while (x <= 2n) x = x + i;
    }
}
```

Note: $H_n = 1 + 1/2 + 1/3 + \dots 1/n$ is called harmonic number and it is known that $H_n = O(\ln n)$.

Exercise 2 [30 points]

a. Let $p(n) = a_m n^m + a_{m-1} n^{m-1} + \dots + a_1 n + a_0$ be a polynomial with degree $m > 0$, where m is an even number. Assume that for every k, $0 \leq k \leq m$, the following hold: (1) if k is even, then $a_k > 0$, and, (2) if k is odd., then $a_k < 0$. Prove that $p(n) = \Theta(n^m)$. [10P]

b. Fill in the brackets ([ ]) the asymptotic order (O, Ω, Θ) of the following 1. expressions (i.e. choose between O, Ω, Θ, so that the expression holds):

[12P]

(i) $n \log \log n$ = [ ] $(n \log^2 n)$

(ii) $\sqrt[3]{n \log n}$ = [ ] $(n^{0.3})$

(iii) $n \log(\sqrt[6]{n})$ = [ ] $(n \log n)$

c. Analyze the asymptotic order (in Θ notation) of the function: [8P]

$$f(n) = n^3 \log n^3 + n^2 \sqrt{n} - n^2 \log n - 1$$

Exercise 3 [25 points]

Consider the following algorithm:

```
int ProcessString(char A[], char B[]) {        // A and B are strings where A has
                                               // greater size than B.
      int n = length(A), m = length(B);        // length() is a helpful function which returns
                                               // the length of the string.
      for (i = 1; i = n-m +1; i++) {
            j = 1;
            while (j <= m && A[i+j-1] == B[j])  j++;
            if (j == m+1) return i;
      }
      return -1;
}
```

a. Present a description of the way the algorithm operates. What is the goal of the algorithm (what is the problem that it solves)?

[5P]

b. Trace the execution of the algorithm when the following strings are provided as its input:

A[1,8] = banenana and B[1,3] = ana. [7P]

c. Analyze the time complexity of the algorithm and its order. Notice that the time complexity of the algorithm will be function of n,m. [13P]

Exercise 4 [25 points]

a. Consider the recursive equation $T(n) = T(n-a) + n + c$. Solve the recursive equation with the method of the repetitive replacement and analyze its order. [10P]

b. Draw the recursive tree for the recursive equation $T(n) = 3T(n/2) + n$. Then, analyze its order using the method of mathematical induction. [15P]