Timothy Hoang

3/31/19

CMPT435-111

Assignment 7

**Date Assigned: 03/25/2019**

**Due: Midnight 03/31/2019 on iLearn**

**Please read turn-in checklist at the end of this document before you start doing exercises.**

**Section 1: Pen-and-paper Exercises**

1. Input: A list of alphabetic characters.

Output: Write a recursive algorithm that determines whether or not the string “yes” occurs in X, and if so, to return the location of the rightmost occurrence of “yes”.

For example, given {'a', 'y', 'e', 's', 'w', 'y', 'e', 's', 'v', 'h', 'e'}, output: 5. Given {'o', 't', 'z', 'y', 'e', 's', 'v', 'g', 'r', 'a', 'y', 'e', 's'}, output: 10

Design a recursive algorithm to solve this problem (15 points).

1. **describe the idea behind your algorithm in English (5 points);**

**(ii) provide pseudocode (5 points);**

**(iii) analyze its running time (5 points).**

**Full credit (15 points) will be awarded for a recursive algorithm. Algorithms that are NOT recursive will be scored out of 5 points.**

1. Analyze the following code and provide a "Big-O" estimate of its running time in terms of n. Explain your analysis.

function fun1(int n)

{

if (n <= 0)

return 0;

else

return n + fun1(n-1);

}

**Note: Credit will not be given only for answers - show all your work:**

**(5 points) steps you took to get your answer.**

**//1 for base case**

**//n for general case**

**(5 points) your answer.**

**Big O(n)**

1. Given an integer n, create a recursive method to find the sum of the series 1^1 + 2^2 + 3^3 + ….. + n^n.
2. **describe the idea behind your algorithm in English (5 points);**

**Base case would be if n = 1 then return 1.**

**General case would be power function of n^n + recursive case of n-1.**

**(ii) provide pseudocode (5 points);**

**if (n == 1) {**

**return 1;**

**}**

**else {**

**return n^n + sum(n - 1);**

**}**

**(iii) analyze its running time (5 points).**

**//1 for base case**

**//n for general case**

**T(1) = O(1)**

**T(n) = 1 + O(n) + T(n-1)**

**Reduce**

**T(n) = n + T(n-1)**

**Replace**

**Big O(n)**

**Full credit (15 points) will be awarded for a recursive algorithm. Algorithms that are NOT recursive will be scored out of 5 points.**

**Section 2: Java Implementation**

1. Implement problem 1 in Java (30 points).

Note:

Find a file called Problem1.java in assignment 3 folder.

Complete the method of where().

Test your method in the main method provided following the comments.

**Full credit (30 points) will be awarded for a recursive algorithm. Algorithms that are NOT recursive will be scored out of 10 points.**

1. Implement problem 3 in Java (30 points).

Note:

Find a file called Problem3.java in assignment 3 folder.

Complete the method of sum().

Test your method in the main method provided following the comments.

**Full credit (30 points) will be awarded for a recursive algorithm. Algorithms that are NOT recursive will be scored out of 10 points.**

**TURN-IN CHECKLIST:**

1. **Answers to Section 1 (.doc/.txt), and to Section 2 (all your source Code (.java files)). Remember to include your name, the date, and the course number in comments near the beginning of your code/report.**
2. **Create a folder and name it 'FirstName\_LastName\_assignment\_3'. In the newly created folder copy and paste your files (.doc/.txt/.java files). Then compress the folder, and submit to iLearn.**