Timothy Hoang

4/28/19

CMPT435-111

Assignment 11

**Date Assigned: 04/25/2019**

**Due: Midnight 04/28/2019 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. Given an array A[] consisting 0s, 1s and 2s, write a function that sorts A[]. The functions should put all 0s first, then all 1s and all 2s in last.

Example:

Input : {0, 1, 2, 0, 1, 2}

Output : {0, 0, 1, 1, 2, 2}

Input : {0, 1, 1, 0, 1, 2, 1, 2, 0, 0, 0, 1}

Output : {0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2}

Design an **in-place O(n)** algorithm to solve this problem (20 points).

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

Make a three way partition that scans the elements and separates them based on wether they are a low or high value. When the reader scans a low value, it swaps out the low value and moves it and the reader up one. If it scans a medium value, it moves the reader up one. If it scans a high value, it swaps the high and reader value and moves the high back a spot.

(ii) provide pseudocode (10 points);

If (A[reader] = 0){

swap A[low], A[reader];

low ++;

reader++

}

If (A[reader] = 1){

reader++

}

If (A[reader] = 2){

swap A[reader], A[high];

high —;

}

End when reader and high meet each other

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

Scans every element once at worst case. 1xn = O(n)

Full credit (20 points) will be awarded for an **in-place O(n)** algorithm. Algorithms that are NOT in-place or slower will be scored out of 10 points.

**Section 2: Java Implementation**

1. Implement Problem 1 in Java.

Note:

Find a file called LinkedList.java in assignment 11 folder.

Complete the method of Sort3way().

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

**Important: In all of the assignments of this course, when you are asked to implement an algorithm for a problem, your code will be evaluated based on:**

**5 points - Execution**

**Each file must run without error or warning on valid input described in the main method provided.**

**5 points - Within Code Documentation**

**Is the code documented for obvious understanding of the use, preconditions, and postconditions of each function?**

**20 points - Correctness**

**Is the algorithm implemented correctly? Does your method pass the test?**

**Note: Full credit (30 points) will be awarded for an algorithm that is O(nlog n). Algorithms that are O(n^2) slower will be scored out of 10 points.**

**TURN-IN CHECKLIST:**

1. **Answers to Section 1 (.doc/.txt/.pdf), 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\_10'. In the newly created folder copy and paste your files (.doc/.txt/.java files). Then compress the folder, and push it to iLearn.**