Assignment 3

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

1. Let L be a list of numbers in non-decreasing order, and x be a given number. Describe an algorithm that counts the number of elements in L whose values are x (English description 5 points + Pseudocode 5 points). For example, if L = {1.3, 2.1, 2.1, 2.1, 2.1, 6.7, 7.5, 7.5, 8.6, 9.0} and x = 2.1 then the output of your algorithm should be 4. **Your algorithm should run in O(log n) time** (10 points).

**Important: In all of the assignments of this course, when you are asked to give an**

**algorithm for a problem, you are (unless otherwise indicated) expected to**

**(i) describe the idea behind your algorithm in English (2 points);**

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

**(iii) analyze its running time (3 points).**

**Regarding requirement (iii): Unless otherwise specified, show the steps of your analysis and present your result using big-O.**

**Note: This problem will be discussed in class. Algorithms that are O(n) or slower will be scored out of 3 points.**

1. You have n coins (n may be even or odd) such that n-1 coins are of the same weight and one coin is heavier than the other coins.

You have a balance scale: you can put any number of coins on each side of the scale at one time, and it will tell you if the two sides weigh the same, or which side is lighter if they do not weigh the same.

Outline an algorithm for finding the coin with different weight.

The number of weighings using your algorithm should be O(log n).

**Full credit (8 points) will be awarded for an algorithm that is O(log n). Algorithms that are O(n) or slower will be scored out of 3 points.**

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

**(ii) analyze its running time (3 points).**

**For this problem, you do NOT need to write the pseudocode.**

1. What is the running time of Binary Search if we use it to search for a number in a sorted linked list?

**(3 points) your answer.**

**(5 points) justification for your answer.**

1. Given a positive integer x, find square root of it. If x is not a perfect square, then return floor (round down).

Examples:

Input: x = 4

Output: 2

Input: x = 11

Output: 3

Outline an algorithm for finding square root of x. Expected in O(log n) time.

**Full credit (10 points) will be awarded for an algorithm that is O(log n). Algorithms that are O(n) or slower will be scored out of 3 points.**

**Note: You should NOT use existing functions like math.sqrt() to obtain the square root of x. Create your own function. Solutions that use existing functions will receive 0 points.**

**(i) describe the idea behind your algorithm in English (2 points);**

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

**(iii) analyze its running time (3 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 count().

Test your method in the main method provided.

**Programs that are O(n) or slower will be scored out of 5 points.**

1. Implement problem 4 in Java (30 points).

Note:

Find a file called Problem4.java in assignment 3 folder.

Complete the method of squareroot().

Test your method in the main method provided.

**Programs that are O(n) or slower will be scored out of 5 points.**

**Programs that use existing functions like math.sqrt() will receive 0 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 push it to iLearn.**