Timothy Hoang

3/10/19

CMPT435-111

Assignment 6

**Date Assigned: 03/04/2019**

**Due: Midnight 03/10/2019 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. 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 (20 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 (5 points);**

**(ii) provide pseudocode (10 points);**

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

Binary search the first and last time the number occurs. Assume the indexes are a and b accordingly. The amount of times it appears will be b-a+1.

//first occurrence of x

While(start<=end){

mid = start + end / 2;

if (L[mid] == x && L[mid-1]!=x){

end = mid-1;

}

else if (L[mid] == x && L[mid-1]==x){

return mid;

]

else if(L[mid]> x){

end = mid-1;

}

else{

start = mid+1;

}

}

//last occurrence of x

While(start<=end){

mid = start + end / 2;

if (L[mid] == x && L[mid+1]!=x){

start = mid+1;

}

else if (L[mid] == x && L[mid+1]==x){

return mid;

]

else if(L[mid]> x){

end = mid-1;

}

else{

start = mid+1;

}

}

Running time is big O log(n) because each is big O log(n) since they are both binary search and 2log(n) simplifies to log(n) when calculating big O.

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 (15 points) will be awarded for an algorithm that is O(log n). Algorithms that are O(n) or slower will be scored out of 5 points.**

**(i) describe the idea behind your algorithm in English (10 points);**

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

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

If n is even, put half the coins on each side. Discard the coins on the side that is lighter and repeat with the coins on the heavier side until there are two coins left then the heavier side has the coin with the different weight.

If n is odd, take one coin out and weigh half of the remaining coins on the scale. If both sides have the same weight, the coin that was taken out in the beginning. Is the heavier coin. If one side is heavier, continue with the same method that we did when n was even.

By doing these methods, you only have to weigh the coins n/2 times at most given us a big O of (logn).

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

**(5 points) your answer.**

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

nlog(n)

N times to get to the middle element each time.

You get to the middle element log(n) times because its the same as binary search.

**Section 2: Java Implementation**

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

Note:

Find a file called Problem1.java in assignment 6 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 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\_5'. In the newly created folder copy and paste your files (.doc/.txt/.java files). Then compress the folder, and push it to iLearn.**