Timothy Hoang

3/03/19

CMPT435-111

Assignment 5

**Date Assigned: 02/25/2019**

**Due: Midnight 03/03/2019 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. Consider the following numerical questions game. In this game, player 1 thinks of an integer in the range 1 to n, where n is an integer. Player 2 has to figure out this number by asking the fewest number of true/false questions.

For example, a question may be “Is your number larger than x?”

Assume that nobody cheats.

1. What is an optimal strategy if n is known? Describe your algorithm (English description 5 points), and Analyze your algorithm’s running time (5 points).

You keep track of the lowest number (starts at 1) and the highest number (starts at n) and guess the average of the numbers. If it is lower, you change the highest number to the current guess - 1 and if it is higher, you change the lowest number to the current guess + 1. The next guess is the new average.

Binary search = big O log(n)

(b) What is a good strategy if n is not known? Describe your algorithm (English description 5 points), and Analyze your algorithm’s running time (5 points).

To find the upper limit of n, you can do the reverse of binary search, also known as exponential, doubling, galloping, or geometric expansion search. You can start with a min of 0 and a max of 1 and see if the number is greater than the max. If it is, then you can set the min to the max plus one because if it’s greater than the max then it is also greater than the min and everything in between. The max can then be multiplied by 2 so it grows exponentially. Once you find a range in which the number is, you continue with normal binary search.

Reverse binary search + binary search = big O 2log(n) = big O log(n)

**Section 2: Java Implementation**

1. Given an access log, create a hashtable using the given SeparateChainingHashTable class to count the number of times an ip address occurs in the access log.

**Note:**

**Find the file named AccessLog.txt, item.java, and SeparateChainingHashTable.java in assignment 5 folder.**

**Complete the main method.**

**Test the main method following the comments.**

**Full credit (30 points) will be awarded for an algorithm that uses SeparateChainingHashTable. Algorithms not using hashtable will be scored out of 10 points.**

**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?**

**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\_5'. In the newly created folder copy and paste your files (.doc/.txt/.java files). Then compress the folder, and push it to iLearn.**