Lab 06 Linear vs Binary Search

Bradley Grose

# Problem

In this lab, we must make a program that tests the difference in binary versus linear search. It must run 20 tests that have an array with 1000 integers between 0-99 and sored using Quick Sort. The program must randomly select a value in that range, and then use both search functions and have counters for the number of times it is found. It then must return if the number is in the array as a Boolean and the number of checks. This is then put into a format for the user to read.

# Solution

For my solution, I did it all in one class. I started by defining static counters for the total as well as individual tests to keep track for reporting results. I also created constants for the array size and test amount. In my main method, I have it print out a welcome message. It then runs a for loop to run 20 tests, in which it uses the for-loop counter to print out a header then run the test. In the test void, it resets in the individual counters. Then it creates a sorted array with 1000. For this array, I used a random number generator to use a for loop to fill the array. Then I used a quick sort algorithm that uses partitions to do a quick so quick sort arranging the numbers in order. This uses mid points and pivot points to swap values around. Once it is sorted, it runs a linear search using a random target. That goes to a class that returns a Boolean if it is found, same goes for the binary search. Both of them keep counts of the number of times the for loops are run. Fir the linear search it checks every spot in array until it finds it in order. For the binary search, it uses the binary tree method of finding the midpoint and seeing if it needs to look above or below for the number, searching less spots to find the number. Then the counters and Booleans are displayed and averages out for the results.

# Implementation Problems Encountered

No problems were encountered in this lab.

# Lab Report Questions

1) The Big Oh complexity for the linear search is much more than that of the binary search. The binary search at most will have a 10-check system, while linear can have up to 1000 checks. 10 is must less checks needed to search for a number compared to 1000 by searching every location in a sorted array.

2) The test does validate my assumptions when looking at the data for each individual search as well as the search needed in each individual case. Using an array size of 1000, it shows the drastic differences needed to find a value and how binary search is much much more efficient and less complex to find the answer.