Lab 07 Report: To Do List

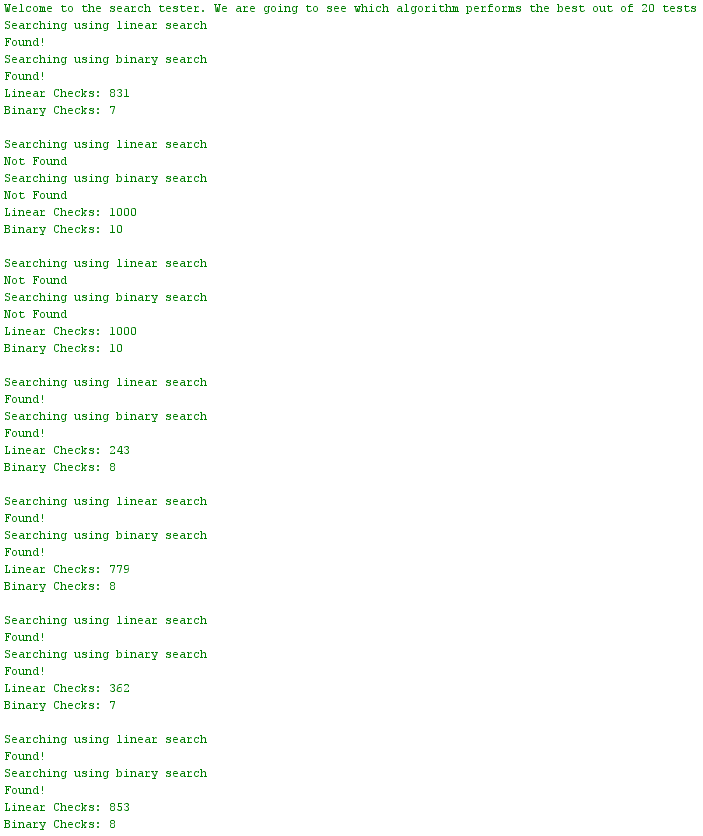
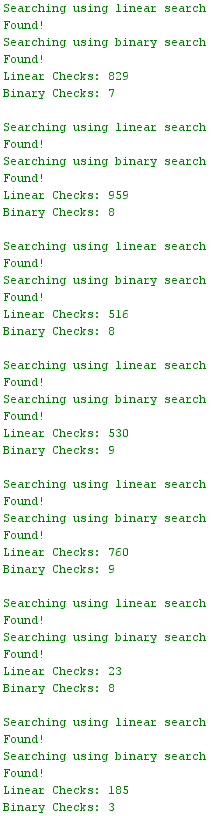
Problem

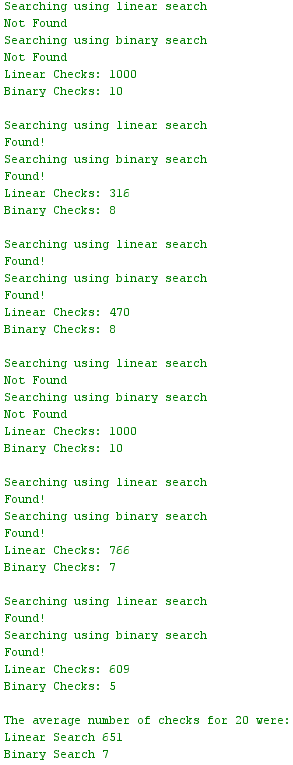
Implement both linear search and binary search, and see which one performs better given an array of 1,000 randomly generated whole numbers (between 0-999), a number picked to search that array at random, and conducting these tests 20 times. Each time the search is conducted the number of checks needs to be counted and at the end the total number of checks should be averaged.

Proposed Solution

1. Create a static method quicksort which also has a private method attached to it which deals with partitioning the left and right sides. This method is what we are using to sort the randomly generated array, even though you can just use Arrays.sort.
2. Create a static method which returns an int array of randomly generated values ranging from 0-999, the array being of size 1000.
3. Create a static method which returns int, the number of checks, named linearSearchRecur which goes through the sorted array find the randomly generated number from the main method. Linear search goes through the array index by index.
4. Create a static method which returns int, the number of checks, named binarySearchRecur which goes through the sorted array to find the randomly generated number from the main method. Binary search breaks the array up into parts based on the index passed in, breaking the array into even more sections until it finds the number.
5. Create a main method which will drive the program. It needs to keep track of how many times each type of search occurs out of 20 tests and calculates there averages. It will print to the console on what is occurring (ie looping the tests 20 times) and finishing off with the average results.

Tests and Results



Problems Encountered

The only problem I encountered this lab was calling the generateArray() method in both calls for the searches which sent in a new array each time. This was fixed by simply creating a int[] temp value which would store the result of the method generateArray() and would then be sent in the method calls for the searches.

Conclusions and Discussion

This lab was straight forward and based on examples and code which was shown in class. This gets students prepared and familiar with these searches and their functionalities. Arrays.sort could have been used instead of implementing quick sort and it’s child method partition within the code as they have the exact same functionality.

Additional Questions

1. What is the Big Oh complexity for linear search? Binary search?

O(n), O(log(n))

1. Plugging in the size of the data in this test into those complexities, did your tests validate those assumptions? Why?

My test validated that binary search would perform faster because it breaks up the array into sections rather than going one by one.