Binary Search Assignment

Introduction:

This report will meet or hopefully exceed the requirements listed and outlined in the given rubric for the “Binary Search Assignment”. It will talk about a program coded entirely by myself, used to search for phone number entries from an index/array using linear and binary searching methods. It will also answer all six discussion questions and talk a bit more about the program. If you wish to find the program defined and explained step by step, look and the comments included in the program.

Testing:

To test my program to see how it handles multiple situations, I took a variety of situations and tried them out with my program. Some of them included the first and last values within the index, the middle value in the index, random values near the start, middle and end from the index, and finally I used values that were not within the index to make sure my program would deal with it in an appropriate way. I used displays to track the numbers while testing to see how the code coped with my tests. When I tested both searches for the first values the binary search took longer because it had to start from the middle while linear search starts from the first value so it found it immediately. When searching for values near the start binary search beat linear search in terms of efficiency, and it went on to continue to beat it in every other test too. Binary search was extremely efficient but on the downside it is very complicated and especially hard to code at my level. I would prefer binary search if I needed to search for values for a big program, long code or project. While for things like tests I would use linear search just because it is quicker to code and easier to follow. But when using linear search, the steps used to find the value are very high, while when using binary search the step used are very low. I think this occurs because for linear search the steps within the loop are repeated constantly while when using recursion only parts of the method will be used so it will take not as many steps. Also when you are searching you are eliminating more than just one data value at a time making way faster than doing things one at a time.

Discussion Questions:

1. The data file or the index of phone numbers was 1500 lines long, so it had 1500 records.
2. The best case scenario for the linear search would be if the number you are searching for is first on the list. While the best case scenario for the binary search would be if the number you are searching for is in the middle of the list rounded down to the nearest whole number.
3. The worst case scenario for each of the search types would be if for linear of the search value is if the number is not found or does not exist. Other than that the next worst scenario would be if the number was the last value in the index/array. The worst case scenario for the binary search would be if the value the user wishes to search for is not found or does not exist within the index/array. Other than that the next worst scenario would be if the value the user wishes to search for is first or last in the array/index, last value being worse by the first.
4. The average case scenario for each search type was for linear searches it would take a lot more time and comparison steps to complete, while binary searches almost always finished before linear searches while taking less time.
5. I determined the average case scenario by taking the scenario that occurs most often or is most likely to occur. In this case there is no number that is more likely than the other so I took the worst and the best cases and averaged them out.
6. No. It is only more efficient to sort a set of data and search for a value using binary search if you are searching multiple times or you are searching for an amount that is located at the back of the list. Linear searching may be simpler to code and use but it is less efficient in most cases because you have to do multiple steps and repeat them multiple times instead of getting straight to the point.

Conclusion:

In conclusion, binary search is much more efficient than linear search and can be very helpful in the long run and on a large scale, if you don’t mind the inefficiency though then linear search would be much simpler and easier to code.

Sources:

My sources are very limited because I thought about most of these concepts myself. But I did receive useful help from multiple sources. I received most of my help from my brother, he explained binary search in detail for me and outlined the parts I had trouble understanding, he also helped make my code easier to read for other people by critiquing it. The second most useful source would be Mr. Berry’s power point on the types of searches, I believe it is named “Searches”, it outlined some of the parameter’s I needed to use, and explained briefly how the searches work. The third source I found somewhat use full would be some websites from the internet I used to define certain words and check for some classes and errors the following is a list of sites that I used to help me in this assignment.  
<http://en.wikipedia.org/wiki/Best,_worst_and_average_case>

<http://www.roseindia.net/java/string-examples/string-replaceall.shtml>

<http://docs.oracle.com/javase/1.4.2/docs/api/java/lang/StackOverflowError.html>