Trevor Cargile

CS 560

Assignment 2, Problem 5 Analysis

**Program Results**:

The following data displays the average amount of compares that occurred with each different type of search and size of array.

A[500] Binary = 8 compares

A[500] Ternary = 18 compares

A[1000] Binary = 9 compares

A[1000] Ternary = 18 compares

A[2000] Binary = 10 compares

A[2000] Ternary = 21 compares

A[4000] Binary = 11 compares

A[4000] Ternary = 24 compares

A[8000] Binary = 12 compares

A[8000] Ternary = 24 compares

Or in table format..

|  |  |  |
| --- | --- | --- |
| **Array Size (n)** | **Binary Compares** | **Ternary Compares** |
| 500 | 8 | 18 |
| 1000 | 9 | 18 |
| 2000 | 10 | 21 |
| 4000 | 11 | 24 |
| 8000 | 12 | 24 |

**Analysis**:

As you can see from the table above, Ternary search not better than binary search. The number of compares for binary search goes up by one for every multiple of 2n, but the ternary search starts at more comparisons, probably because you need to compare the values to 3 values to figure out what third you need to search through.

**Conclusion:**

If you were to use Binary search or Ternary search, the difference in times would probably not be noticeable, but if you really wanted the most optimal search algorithm, the binary search algorithm would be the best one to do. Ternary has a little more overhead than the binary search, but not too much to differ you away from using it. In the end, the complexity of both binary search and ternary search are logn, but constants could make the ternary search slower than the binary search.