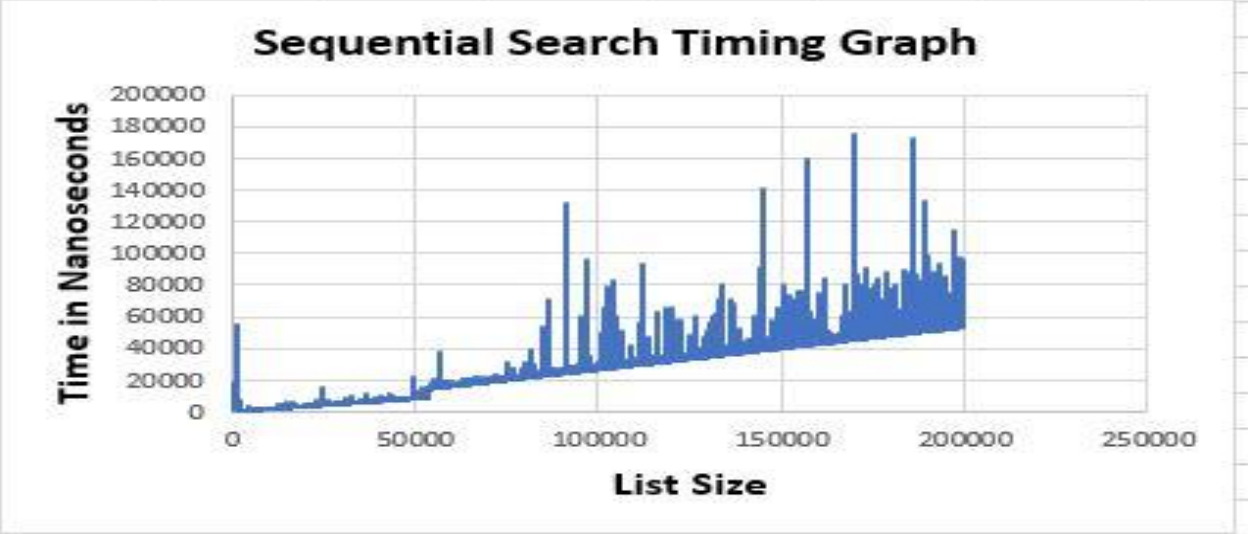
PID: 6169881

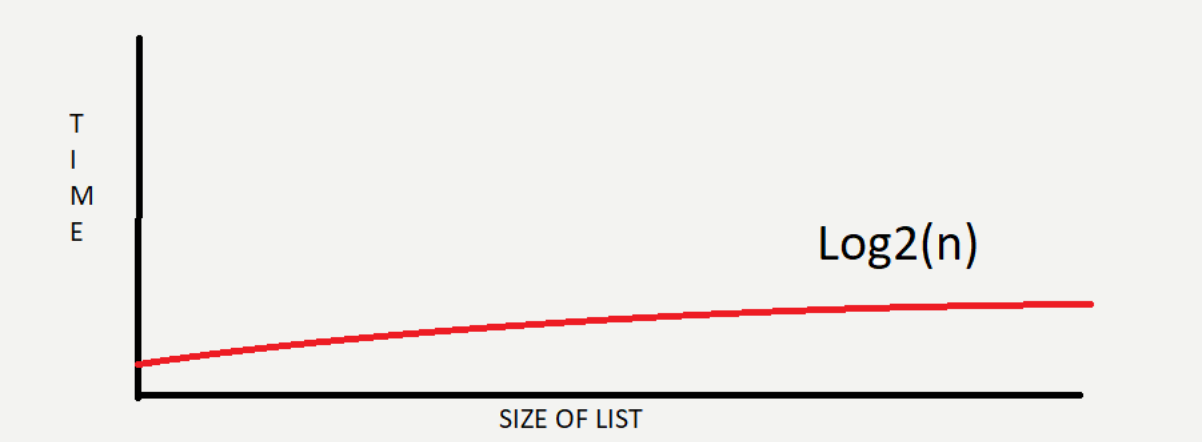

Binary Search Timing Graph
Time in Nanoseconds
30000
25000
20000
15000
10000
5000
0
0
50000
100000
150000
200000
250000
List Size


Sequential Search Timing Graph
Time in Nanoseconds
200000
180000
160000
140000
120000
100000
80000
60000
40000
20000
0
0
50000
100000
150000
200000
250000
List Size


Sorted Search Timing Graph
Time in Nanoseconds
700000
600000
500000
400000
300000
200000
100000
0
0
50000
100000
150000
200000
250000
List Size

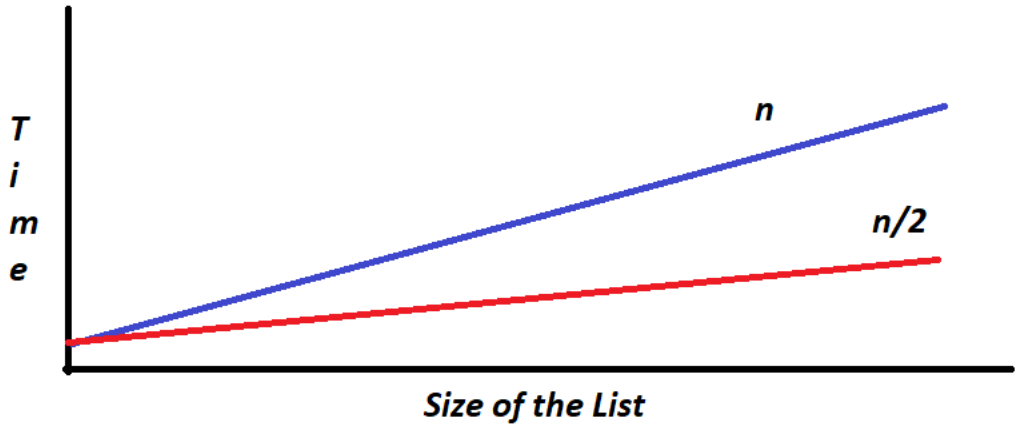
After observing the behavior of the graphs listed previously, I was able to conclude that the fastest searching method among Binary Search, Sequential Search, and sorted search, is Binary search.

Binary search time complexity worst case scenario is **=O(*log$_2$(n)*)**, and the graph for that function is a semi-curved line, that slowly is increasing over time, but tends to be constant the bigger the array size is. The following picture is the graph for ***log$_2$(n)***:



The next more efficient search method is the sorted search. Even though this method uses the regular linear search concept and the worst-case scenario is **=O(*n*),** it gives us some advantages since the array is sorted. A normal linear search will traverse the entire array until the value (x) is found. Let's say we have the following sorted array [1, 2, 3, 5, 6, 7, 8, 9, 10] and x = 4. Once we get to the value of 5 we know that is impossible that 4 is on the rest of array, since it is sorted. In that moment we can stop our loop and return false ***if (i > x) return false.*** On the other hand, if the array was unsorted this condition cannot exist since the number, we are looking for can be the last one. On a sorted array the chances that our number is in the last index are ***(1/n)*** being ***n*** the size of the array. For this reason, the average case scenario is ***(n/2),*** less than the worst-case scenario than the unsorted linear search which is ***=O(n).***

Both graphs for sorted and unsorted sequential search are linear. But the slopes are different.



These graphs represent the average behavior for the graphs in my observations.