Search Test Lab Report

Names: **Nachiket Dani**

**1. Linear Search**

We know from class that the theoretical time complexity of linear search over *unordered lists* is:

|  |  |  |
| --- | --- | --- |
| **Best Case** | **Worst Case** | **Average Case** |
| *1* | *N* | *N/2* |

**Q1:** Increasing the number of trials and the value of N

1. Run experiments with an increasing value of N (from 1000 to 10,000). Does increasing N affect how many trials you have to run to get accurate results? Explain.

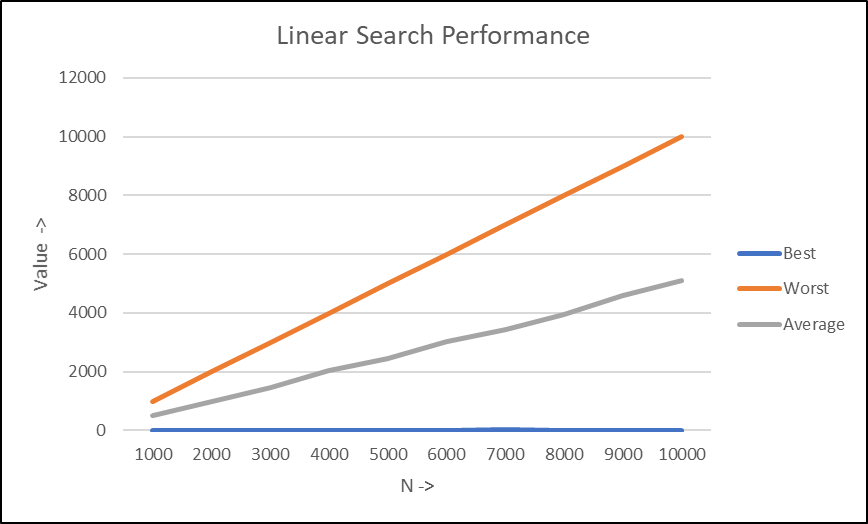
Answer: Increasing N does affect the number of trials needed to run to get accurate results. It is akin to sample size determination. The bigger the population (N) the bigger the sample size (trials) needed to represent the time complexity of the linear search. As the ratio of number of trials to N (list length) approaches 1, the results match closely to the expected behavior (linear) of the linear search algorithm.

1. Write down the number of trials that seem to have worked well for N=10,000.

|  |
| --- |
| **Number of Trials** |
| 1000 |

Number of trials did not really matter after a certain point to improve variance in the best and average performance of the linear search. However, the variance in the average, best and worst outcomes does reduce as the number of trials increases.

**Q2:** Linear Search Time Complexity Plot (Unordered List)

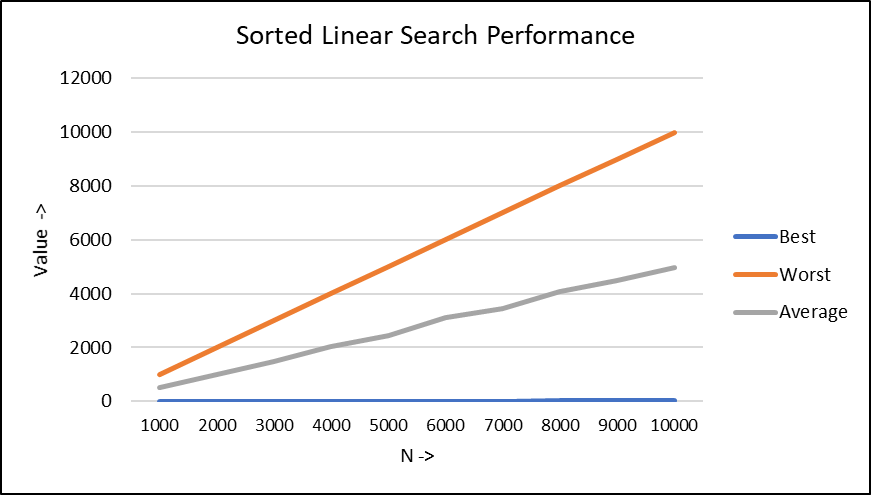


**Q3:** Does the order of the data in the list affect the number of comparisons? In the table below, guess the time complexity of Linear Search on an *Ordered List.*

|  |  |  |
| --- | --- | --- |
| **Best Case** | **Worst Case** | **Average Case** |
| 1 | N | N/2 |

Sorting should not affect the linear search

Linear Search Time Complexity Plot (Ordered List)



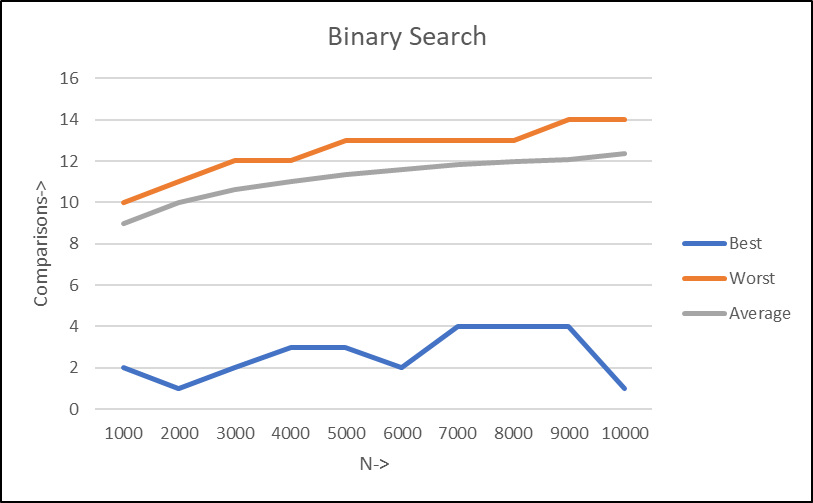
**Conclusion:** The order of the data in the list does not affect the number of comparisons. The plots for best, average and worst case scenarios for sorted and unsorted lists look similar and produce similar results for the number of comparisons.

**2. Binary Search**

We know from class that the theoretical time complexity of binary search over *ordered lists* are:

|  |  |  |
| --- | --- | --- |
| **Best Case** | **Worst Case** | **Average Case** |
| *1* | *log\_2(N)* | *???* |

**Q4:** Binary Search Time Complexity Plot



**Conclusion:** What do your results tell you about the average-case complexity of Binary Search?

Results suggest that the average and worst case complexity are approximately the same. Whereas the best case does not typically follow a pattern. The Average time complexity levels out and looks to be O(log n).

**3. Median**

Q5: We hypothesize that the time complexity of find\_median is:

|  |  |  |
| --- | --- | --- |
| **Best Case** | **Worst Case** | **Average Case** |
| N | N^2 | (N^2) / 2 |

**Justification:**

1. Best case scenario:

*Happens when...*

The first value in the list is the median. In this case the outer loop only executes once (for i = 0). The inner loop still runs through all elements in the list to ensure that the count of less\_than is the same as grt\_than. So if the first value is the median, the algorithm concludes for the first value of i. Best case complexity is O(N).

1. Worst case scenario:

*Happens when...*

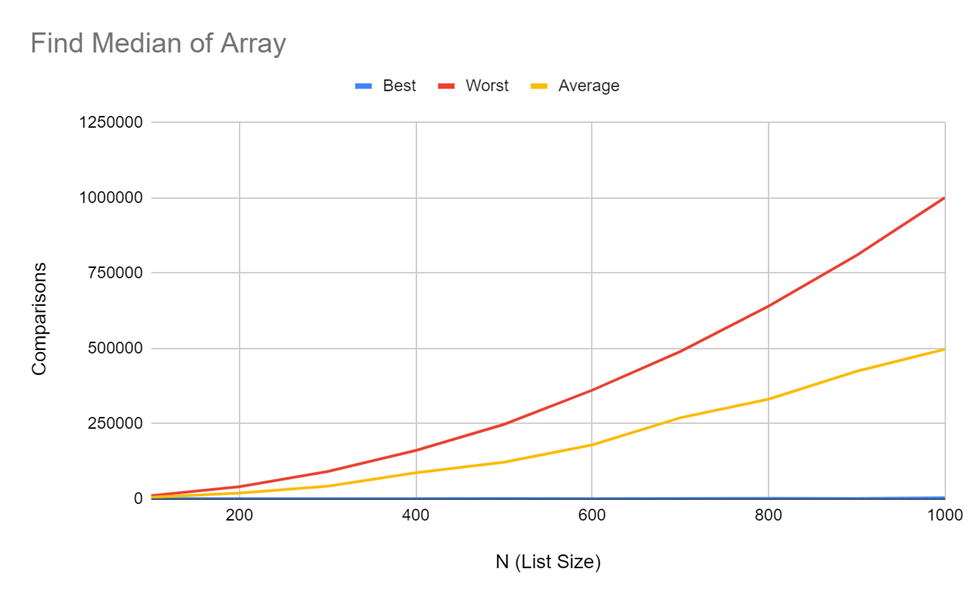
The last value is the median. Contrary to the best case scenario, the outer loop has to iterate through all values (till it reaches the last element) which is only when will we get to a condition that fulfills the algorithm (less\_than == grt\_than for odd and less\_than – grt\_than = 1 for even count list). Worst case complexity = Ω (N\*N)

1. Average case scenario:

*Observed when…*

The median value is located approximately in the middle of the list. Thus, the outer loop runs to roughly N/2 before we are able to find the median value for the list. The inner loop executes for N\*(N/2) times. Thus average case complexity for find median algorithm is O((N^2) / 2)

Find\_median Time Complexity Plot



**Conclusion:** Did your results support your hypothesis? If not, why not, and how does it change your original hypothesis?

The results do support my hypothesis with regards to the “find median”. We ran the code for 500 trials to see the pattern that makes the hypothesis evident. Results would have been even more conclusive if we run more trials.