Search Test Lab Report

Names:

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.

Yes, because as the value of N increase, the probability of picking any element in the list decreases, so we need more trials to ensure that we have a good representation of the algorithm. If we do not have enough trials, we need to end up with results that are not representative of the true performance of the algorithm.

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

|  |
| --- |
| **Number of Trials** |
| 5000 |

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

|  |
| --- |
| *Chart, line chart  Description automatically generated* |

**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 |

Linear Search Time Complexity Plot (Ordered List)

|  |
| --- |
| *Chart, line chart  Description automatically generated* |

**Conclusion:**

The order of the data in the list has no influence on time complexity for linear search.

**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)* | *Log\_2(N)* |

**Q4:** Binary Search Time Complexity Plot

|  |
| --- |
| Chart, line chart  Description automatically generated |

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

Although the average case time complexity is typically smaller than the worst case of time complexity, the difference between them decreases as the input size, N, increases. Therefore, in the case of larger inputs, the time complexity for the average case is similar to the worst case.

**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 median be the first element of the list, then the outer for loop of the algorithm will only run once. As a result, the time complexity in best case scenario can be expressed as N.*

1. Worst case scenario:

*Happens when...* *the median be the first element of the list, then every other element in the list needs to be compared to the first element to determine the median. This means that the algorithm will need to perform N-1 comparisons, resulting in a time complexity of N^2.*

1. Average case scenario:

On average, the median converges to the middle which is (N^2)/2.

Find\_median Time Complexity Plot

|  |
| --- |
| Chart, line chart  Description automatically generated |

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

Yes, the results support the hypothesis, no change.