Search Test Lab Report

Names: Xiaoxia Yin

**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. Increasing the N increases the number of trials to get accurate results. According to the graphs of time complexity, the sample size (N) significantly affects the result of the number of computations for all non-constant growth rates. Therefore, we will need more trials to meet best, worst and average cases.

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)

A screenshot of a map

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)

A screenshot of a map

Description automatically generated

**Conclusion:**

For linear search, the order of the data in the list does not affect 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

A close up of a map

Description automatically generated

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

The average-case complexity of Binary Search is similar to the worst case of it, which is logN

**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 |

\*if we add a comparisons counter before the conditions evaluating whether the list length is odd or even, it would be N+2, (N+2)^2 and (N+2)^2/2

**Justification:**

1. Best case scenario:

*Happens when...*

The list is well ordered by less than and greater than median values so it only needs to run all items once

1. Worst case scenario:

*Happens when...*

The list is reversely ordered by less than and greater than median values so it has to run N\*N times

1. Average case scenario:

When the list is not ordered as the best/worst case scenario

Find\_median Time Complexity Plot

A screenshot of a cell phone

Description automatically generated

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

It supports my hypothesis.