Search Test Lab Report

Names: Chenxi Cai

**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 N needs more trials if I want to get accurate results. The reason is that larger N means more data needed to be passed through during searching, and then the number of comparisons have more different possibilities. If we want to get accurate results, we need to try more times (more trials) to get more possible comparisons. The more possible comparisons, the more accurate results.

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

|  |
| --- |
| **Number of Trials** |
| 1600 |

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

|  |
| --- |
| *Insert plot here* |

**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)

|  |
| --- |
| *Insert plot here* |

**Conclusion:**

The order of the data in the list doesn’t affect the number of comparisons. The number of trials affect the accuracy of results.

**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

|  |
| --- |
| *Insert plot here* |

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

From the plot, I guess that the average-case complexity of binary search is *log\_2(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 element in the list is the median.*

1. Best case scenario:

*Happens when the last element in the list is the median.*

1. Average case scenario: The element in the middle position of the list is the median.

Find\_median Time Complexity Plot

|  |
| --- |
| *Insert plot here* |

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

Yes. My results support my hypothiesis.