Search Test Lab Report

Names: Weisong Chen

**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 affect number of trials we need to run to get accurate results, especially affecting the best case and worst case scenario.

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

|  |
| --- |
| **Number of Trials** |
| 2500 |

**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 |

Linear Search Time Complexity Plot (Ordered List)

**Conclusion:**

Order of the list does not affect the time complexity of Linear Search.

**2. Binary Search**

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

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

**Q4:** Binary Search Time Complexity Plot

|  |
| --- |
|  |

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

Average-case complexity is closer to worst case scenario than to best case scenario. It is highly likely that the average case complexity of Binary Search is also O(logn).

**3. Median**

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

|  |  |  |
| --- | --- | --- |
| **Best Case** | **Worst Case** | **Average Case** |
| n | n2 | About n2/2 |

**Justification:**

1. Best case scenario:

*Happens when the median is the first element of the array, in which case we just have to scan through the rest of n-1 element.*

1. Worst case scenario:

*Happens when the median is the last element of the array, in which case for each element in the array, we have to compare it with all other rest elements.*

1. Average case scenario:

*The median is in the middle somewhere in the array.*

Find\_median Time Complexity Plot

|  |
| --- |
|  |

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

Yes, the result support my hypothesis.