Search Test Lab Report

Names: Ha “Kelly” Tran

**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 the larger N is, we need more trials to make the probability of picking each element in the list approaching 1/N. This will increase the accuracy of the results.

For example, for N=10, num\_trials = 100, the probability of picking an element will be close to 1/10.

Meanwhile, for N=10,000, num\_trials = 100, the probability of picking 9,900 of the elements (99% of the elements) will be 0. The probability of picking the key can be 0 in this case.

Hence, the number of trials need to increase as N increases for the results to be more accurate.

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)

|  |
| --- |
|  |

**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.*

*No,* the order of the data in the list does not affect the number of comparisons

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

Linear Search Time Complexity Plot (Ordered List)

|  |
| --- |
|  |

**Conclusion:** The order of the data in the list does not affect time complexity for linear search (the key is randomized).

**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

|  |
| --- |
|  |

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

The time complexity for average case is similar to that of the worst case. The average case time complexity is generally smaller than the worst case time complexity, but as N gets larger, the difference gets smaller.

**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 is the first element of the list. The outer for loop runs once. Hence, the time complexity for best case is N.*

1. Worst case scenario:

*Happens when...* *the median is the first element of the list. Every indexes need to be checked. Hence, the time complexity for worst case is N^2.*

1. Average case scenario: Since the list is unordered, the median can be anywhere in the list. On average, the median converges to the middle

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 data above runs on 500 trials, and the result supports the hypothesis. No change was needed.