Search Test Lab Report

Names: Emiliano Zhu

**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.

Answer: Yes. The greater the input size N is, the greater the random error will be. Therefore, we need to run more trials in order to get accurate results.

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.*

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

Linear Search Time Complexity Plot (Ordered List)

|  |
| --- |
|  |

**Conclusion:**

* The time complexity of linear search over unordered list is:
  + Best Case: *1*
  + Worst Case: *N*
  + Average Case: *N/2*
* The time complexity of linear search over ordered list is:
  + Best Case: *1*
  + Worst Case: *N*
  + Average Case: *N/2*

**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) - 1* |

**Q4:** Binary Search Time Complexity Plot

|  |
| --- |
|  |

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

* The time complexity of binary search over ordered list is:
  + Best Case: *1*
  + Worst Case: *log\_2(N)*
  + Average Case: *log\_2(N) - 1*

**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 number of the input list is the median of the list.*

1. Best case scenario:

*Happens when the last number of the input list is the median of the list.*

1. Average case scenario:

*O((1\*N + 2\*N + … + N\*N) / 2) = O(N^2/2)*

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, my results support my hypothesis. The time complexity of find\_median over unordered list is:
  + Best Case: *1*
  + Worst Case: *N^2*
  + Average Case: *N^2/2*