Search Test Lab Report

Names: Yiyu Zhang

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*

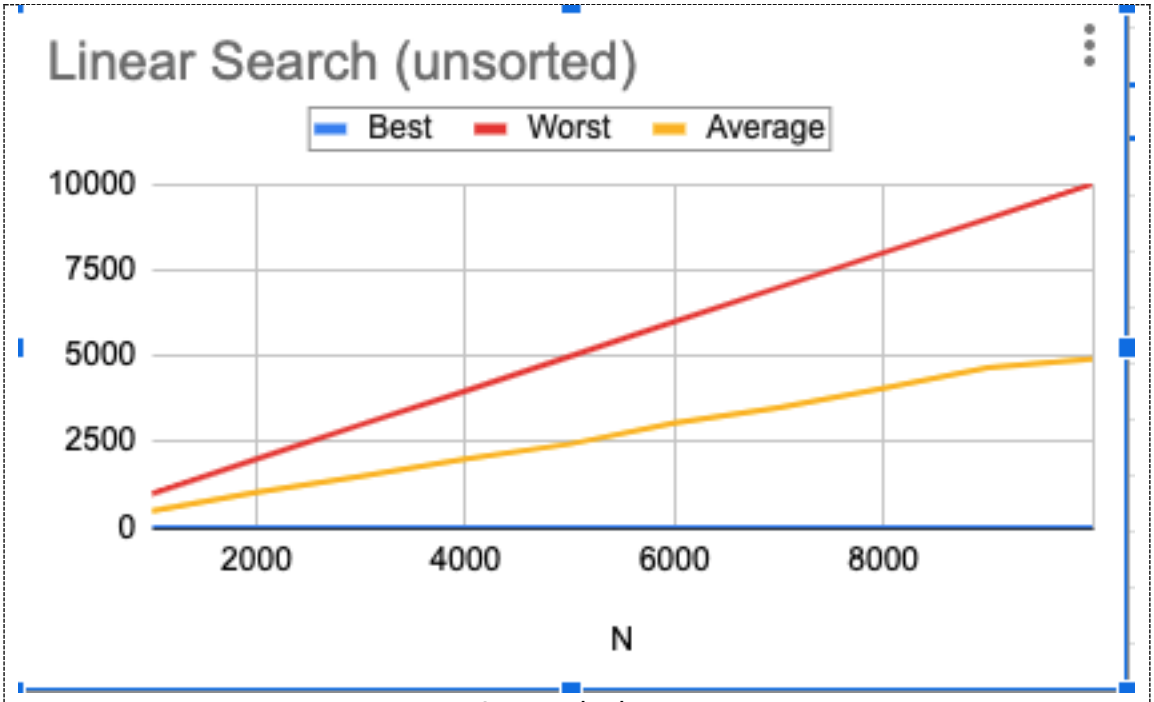
Q1: Increasing the number of trials and the value of N

A. 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 how many trials have to run to get accurate results. I've tried running the linear search with fixed number of trials(trials = 200), and also running the linear search with increasing number of trials as N increased, and the result shows that when running the search with increasing number of trials as N increased, the difference between average number and expected average number are smaller. That's because we have to increase the sample size(number of trials) to lower the variance.

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

Number of Trials
8000

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:

When performing linear search, whenever the input list is sorted or unsorted, the best case scenario takes O(1) time to search for the target element, and takes O(n) time in the worst cases. The average of the time complexity is O(n/2), and can simplified as O(n).

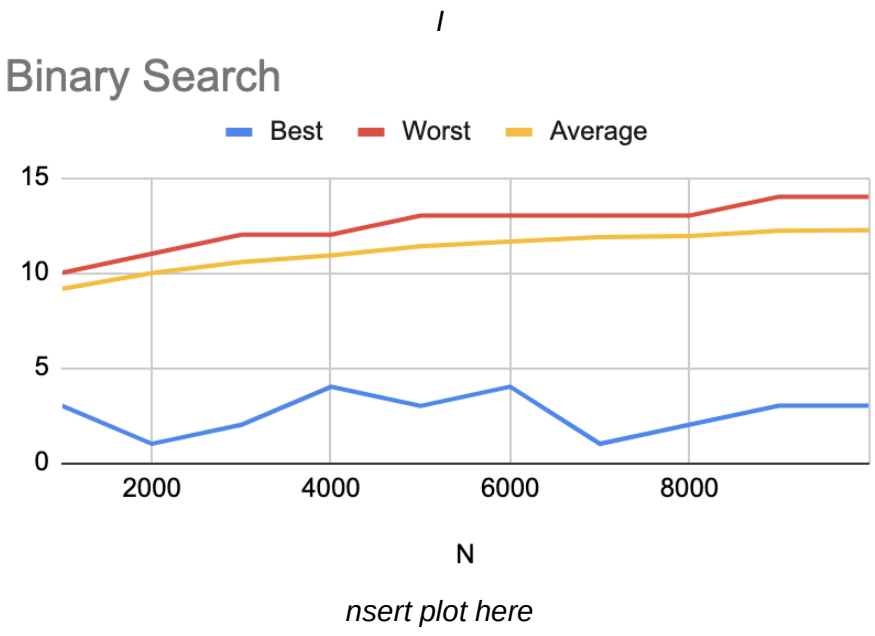
2. Binary Search

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

I



nsert plot here

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

The average-case complexity of Binary search is the same as the worst case complexity.

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:

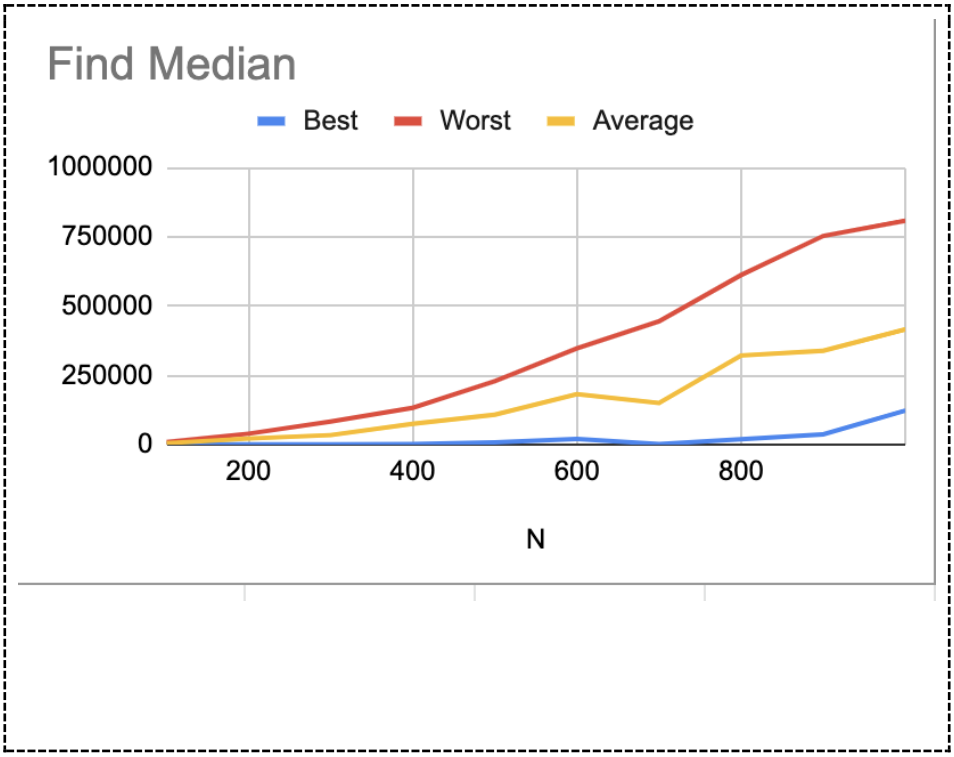
A. Best case scenario:
Happens when the list has odd numbers of elements and less_than == grt_than or, the list has even numbers of elements and less_than == grt_than – 1 in the first outer for loop.

B. Worst case scenario:
Happens when the list has odd numbers of elements and less_than == grt_than or, the list has even numbers of elements and less_than == grt_than – 1 in the last outer for loop.

C. Average case scenario:
Happens when the median number is in the middle of the list.

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?

My result support my hypothesis.