PROJECT 2

Name: Yashwanth Raj Varadharajan
Class: CS6212
GWID: G47635180

Problem Statement:

We are required to analyze Quickselect algorithm Median of Medians method to find the k^{th} smallest element in any given array.

Theoretical Analysis

Traditional way of solving the problem is to sort the array and select the k^{th} element from the start. But the sorting algorithm will take O(nlog(n)) time to complete the sorting. We use Quickselect algorithm where we can find the k^{th} smallest element without sorting, and Time complexity of this algorithm in **O(n).**
In this algorithm, we select a pivot element and use it to create smaller subarrays. The arrays are broken into smaller arrays of length 5, and the median of each smaller array is calculated. Then we analyze the medians of all the smaller arrays and recursively use it to find the median of medians, which we use as pivot element. This way we can effectively ensure that the pivot is close to true median of given array.

Algorithm Analysis

- The algorithm starts by checking the size of the array. If less than 5, then it sorts the array and returns the k^{th} element.
- If size of arrays is larger than 5, then the algorithm divides the array into groups of 5 elements. Then it calculates the median of all the groups.
- After processing all the groups, it calculates the median of all the medians from smaller groups.
- Now, the median of medians is used as the pivot element to partition the array into three parts, Ones that are less than pivot, Ones that are equal to pivot, and Ones that are greater than pivot.
- Now depending on the given element k, the program makes recursive calls to select one of the 3 divisions created by pivot element.
- After division selection and processing, the algorithm returns the result.

Experimental Analysis

Code : https://github.com/YashwanthRaj/QuickSelect-Algorithm.git

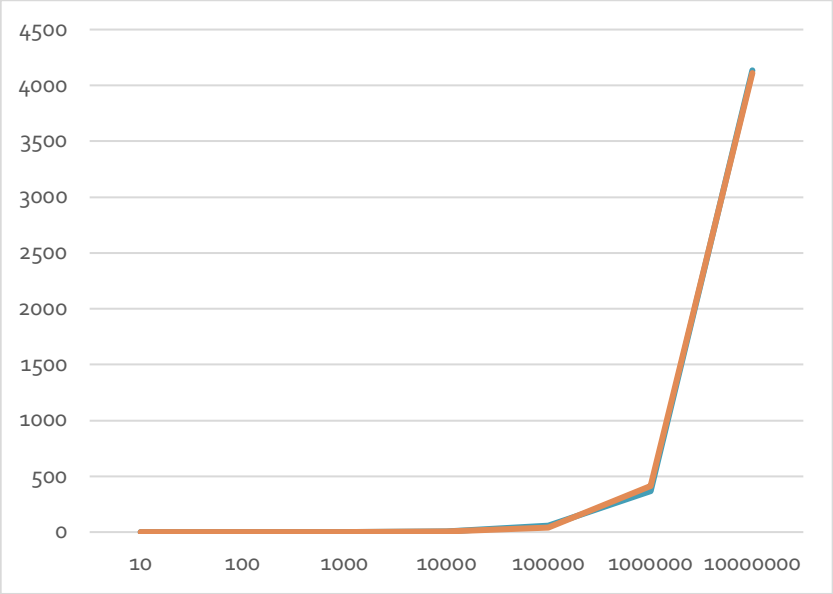
We have formulated and run the code for different values of n ranging from 10 to 10^7, and noted down the time taken for each n value. We also substitute the values of n in time complexity expression that we have derived and tabulate all the results.
Now we notice that the experimental time is in milliseconds, but the theoretical time are constants, hence, to plot graphs, we need to multiply all the theoretical values by scaling constant.

$$Scaling\ Constant = \frac{Average\ of\ Experimental\ Results\ in\ ns}{Average\ of\ Theoretical\ Results}$$

The average of experimental values is 652.4185714 and the average of theoretical values is 1587301.43. Hence dividing them, we get the scaling constant as 0.000411, which we multiply with theoretical values to get the adjusted values that we can plot.

Comparing Experimental and Theoretical values

n	Experimental Values (ms)	Theoretical Values	Scaling Constant	Adjusted Theoretical Constant
10	0.03	10	0.000411	0.00411024
100	0.11	100	0.000411	0.04110237
1000	0.58	1000	0.000411	0.41102374
10000	4.92	10000	0.000411	4.11023741
100000	57.55	100000	0.000411	41.1023741
1000000	369.16	1000000	0.000411	411.023741
10000000	4134.58	10000000	0.000411	4110.23741



Graph Observation:
From the graph we can observe that the theoretical calculation and the experimental calculations are very close to each other. Both the graphs are increasing in linear time O(n).

Conclusion

We have analyzed Quickselect algorithm Median of Medians method to find the k^{th} smallest element in any given array. We have formulated the code and compared experimental and theoretical values.