ASSIGNMENT ONE – MEDIAN ANALYSIS

Name	**Date: 01/11/2024**
▪ محمد شريف فتحى يوسف جليله	ID: 21011151

1. Problem Statement

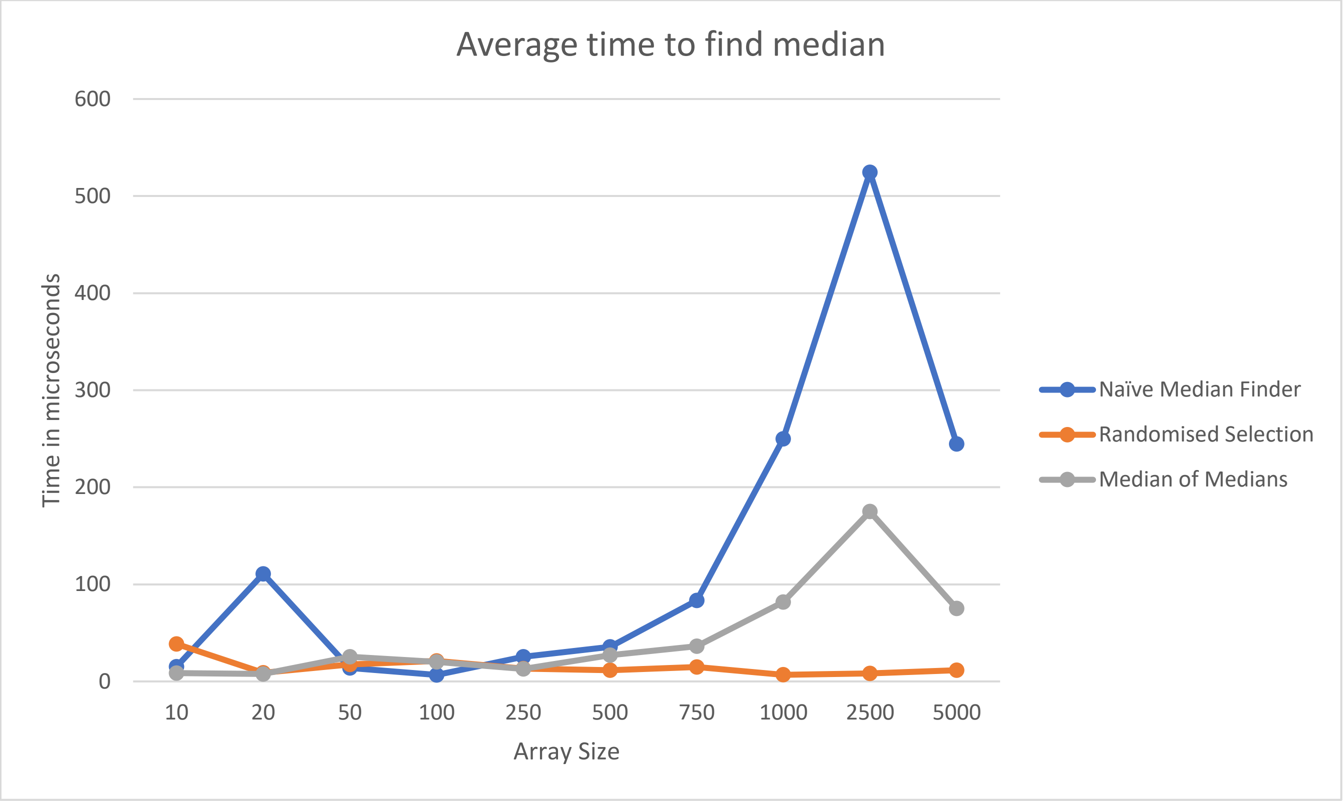
The purpose of this project is to study the time complexity and performance of different median-finding algorithms on large datasets. Understanding the differences in execution time between these methods is crucial in selecting an appropriate algorithm for use cases involving large-scale data processing.

- **Randomized Selection**: This method leverages random partitioning and has an average time complexity of O(n). The algorithm selects a pivot at random and recursively reduces the search space.
- **Deterministic Linear-Time Selection (Median-of-Medians)**: This method ensures a guaranteed O(n) runtime by using the median-of-medians technique, which helps avoid poor performance on adversarial inputs.
- **Naïve Median (Sorting)**: This approach sorts the array and retrieves the median as the k-th smallest element. It serves as a simple baseline for performance comparison with O(n log n) complexity.

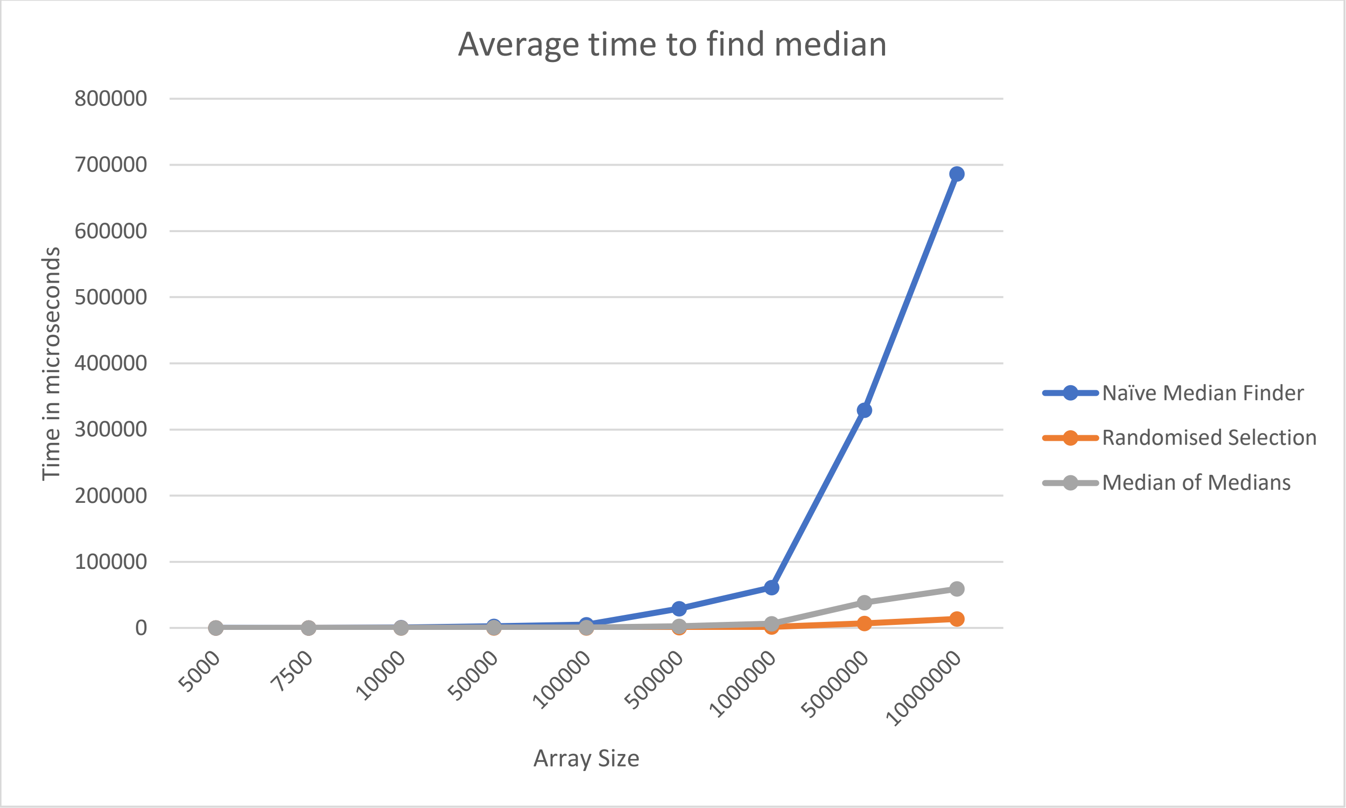
To analyse the performance of the implementations of the algorithms above, a random array generator method, that takes as an input the array size, was used. The three implementations were tested using array sizes 10, 20, 50, 100, 250, 500, 750, 1000, 2500, 5000, 7500, 10,000, 50,000, 100,000, 500,000, 1,000,000, 5,000,000 and 10,000,000. For each array size, the median was found for 10 different arrays. The average of the times for finding the median was then found and the graphs were constructed.

2. Performance Analysis

1) Array Sizes (10 – 5000)



2) Array Sizes (5000 – 10,000,000)



3. Assumptions

As per CLRS 9.2, “the expected running time of RANDOMIZED-SELECT is $\Theta(n)$, assuming that the elements are distinct”, and I, therefore, made it so that the random array generator ensures all elements are distinct and are ranging from 1 to 1,000,000,000 to ensure a greater variety of elements and that the range is much larger than the upper array size bound, which is 10,000,000.

4. Conclusion and Remarks

As array size increases, the naïve method becomes almost unusable, and takes too long to calculate the median, even if it sorts the array in $O(n \log n)$. However, in smaller arrays, with sizes less than 200, the naïve method can generally be faster than randomized selection and deterministic selection, which can be seen through the graphs above. Randomised selection has proven its dominion, and that it is generally better than deterministic selection.

5. Acknowledgement

I acknowledge that I am aware of the academic integrity guidelines of this course, and that I worked on this assignment independently without any unauthorized help.