Order Statistic Lab
COSC 3523 – Analysis of Algorithms
Carter Hidalgo
Ethan Templeton
10/18/2023

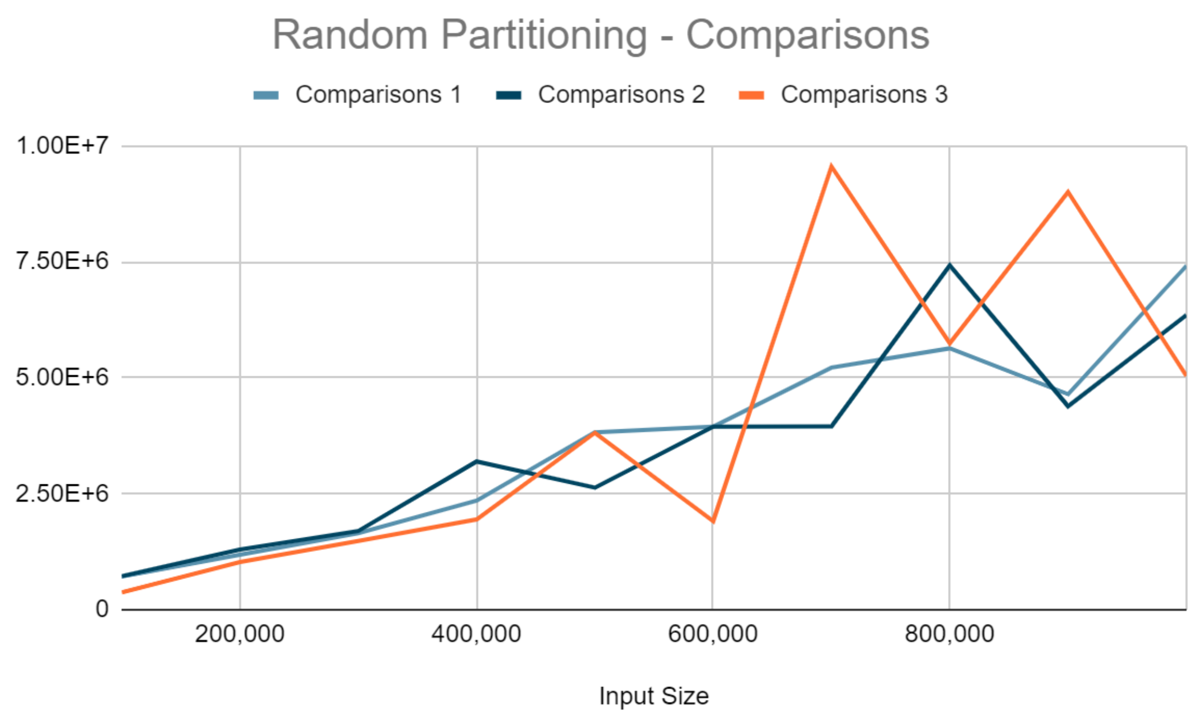
Introduction

This report will provide an empirical analysis of the ith order statistic problem using random partitioning and the median of five solutions. The analysis considers the number of comparisons needed to find a randomly selected ith order statistic in linearly increasing arrays of randomly generated large integers ranging from zero to the maximum integer value. Each solution receives an identical copy of the current array as the array size and values increase and change respectively. Results are shown below in both graph and table form as well as an analysis of the results and an estimate on the order of complexity based solely from the number of comparisons. Source code is provided in the project zip folder and is also available on GitHub at the following URL:

https://github.com/CarterHidalgo/COSC3523__OrderStatisticLab__Java.git

Random Partitioning

Input Size	**Comparisons 1**	**Comparisons 2**	**Comparisons 3**
100,000	712271	722477	372963
200,000	1184135	1295987	1029017
300,000	1650433	1696889	1480101
400,000	2351751	3197171	1946483
500,000	3825913	2631383	3813525
600,000	3944529	3945761	1909053
700,000	5220361	3952443	9554767
800,000	5634159	7426029	5744941
900,000	4646573	4385603	9005475
1,000,000	7413591	6352643	5040357



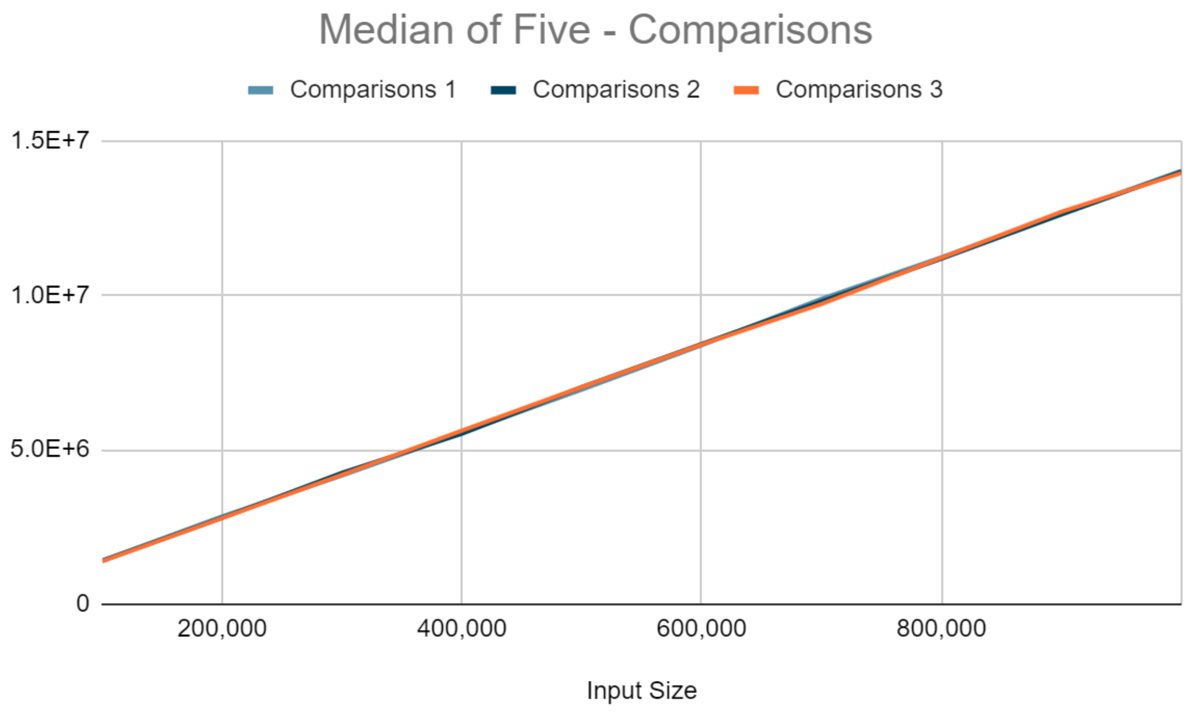
Analysis of Random Partitioning

As you can see, the order of complexity of the algorithm is bounded random, so although it does follow the general expected complexity of E(O(n)), as the input size increases the difference between the best case time and the worst case time grows. The graph bounces between those asymptotes, growing more variable as input size increases. Despite this random variability, tests confirm that this random partitioning algorithm does stay bounded between linear and quadratic time, and that it is more likely to resemble linear.

Median of Five

Input Size	**Comparisons 1**	**Comparisons 2**	**Comparisons 3**
100,000	1419654	1411884	1394201
200,000	2852111	2796696	2790015
300,000	4170970	4258579	4191034

400,000	5549382	5519291	5621761
500,000	6948898	7034399	7028896
600,000	8386611	8425578	8411964
700,000	9888847	9809616	9722625
800,000	11234831	11180047	11213737
900,000	12642206	12611604	12704188
1,000,000	14030769	14006501	13955021



Analysis of Median of Five

The median of five solution demonstrates a remarkable level of consistency. Across three test runs of ten array sizes each, the algorithm consistently has a linearly increasing run time leading to a practical time complexity of O(n). The median of five algorithm guarantees a good split when partitioning the array by finding the median of medians to partition the input array around. While this does lead to a larger number of comparisons relative to the random partitioning approach, the benefit is that the runtime will never be worse than O(n). In simple terms, the median of five guarantees O(n) complexity in every instance by making extra comparisons upfront.

Conclusion

Both the random partitioning solution and the median of five solution provide fast methods for finding the ith order statistic. This report has shown that while the random partition solution is practically O(n), it still has the potential for a much larger O(n^2) complexity if a bad partition is found. Conversely, the median of five has the tradeoff of more comparisons for the guarantee of a good split. If reliability is needed, the median of five approach is probably best, since it is by far more consistent than random partitioning. If reliability is not a concern or the number of comparisons or run time is important, then random partitioning is probably the better approach.