Luke Pepin

CSE 3500

HW7 - Quicksort

1. Table comparing running time (Average of 5 runs):

| **Algorithm** | **Input (Time in Seconds)** | | | | |
| --- | --- | --- | --- | --- | --- |
| 500 | 800 | 1000 | 1200 | 1500 |
| Sorted-Mid\_mids | 0.212 | 0.254 | 0.268 | *0.266* | 0.277 |
| Sorted-Last | 0.254 | 0.279 | 0.296 | 0.324 | 0.364 |
| Random-Mid\_mids | 0.235 | 0.259 | 0.263 | 0.272 | 0.277 |
| Random-Last | 0.276 | *0.252* | 0.281 | 0.286 | *0.284* |

\**Italics: Outlier*

1. Conclusion:

While using the last element as a pivot for Quicksort is simple to implement, it can often lead to worst case scenarios with poor performance. It is particularly striking the Sorted-Last Algorithm where the running time for the larger data sizes increased drastically. While in both the Sorted and Unsorted arrays the median of medians provided a much more consistent result and predictable information (Attempting to maintain its O(n log n) time complexity). As a result of this experiment the median of medians is a much more preferable pivot to the last element pivot as it is much more consistent in its running times.

1. Source Code attached with submission QuickLast.py and QuickMed.py