# Analysis of Quick Sort Variants | Lab 7

# Experiment Design

In order to analyse performance of each quicksort variant, I ran each algorithm with increasing sized data sets with ascending, descending and randomly ordered data. These datasets were generated using the *gen* program, provided in last week’s lab. The sizes of each dataset (excluding the first 3 sizes), were of base two. This was so we could easily deduce the time complexity based on their relative time increments. The order in which we ran these experiments are as follows:

* For order in Ascending, Random, Descending
  + For size in 100, 1000, 2000, 4000, 8000, 160000
    - Run / Time sorting algorithm

Each algorithm run was repeated 5 times with their results averaged.

# Experiment Results

The results below have been collected after running the experiments, as described above.

## Quicksort using Naïve Pivot

|  |  |  |  |
| --- | --- | --- | --- |
| Items | Ascending | Random | Descending |
| 100 | 0.000 | 0.000 | 0.000 |
| 1000 | 0.000 | 0.000 | 0.000 |
| 10000 | 0.132 | 0.000 | 0.140 |
| 20000 | 0.540 | 0.000 | 0.560 |
| 40000 | 2.184 | 0.010 | 2.256 |
| 80000 | 8.752 | 0.020 | 9.010 |
| 160000 | 34.968 | 0.040 | 36.058 |

## Quicksort using Median of Three Pivot

|  |  |  |  |
| --- | --- | --- | --- |
| Items | Ascending | Random | Descending |
| 100 | 0.000 | 0.000 | 0.000 |
| 1000 | 0.000 | 0.000 | 0.000 |
| 10000 | 0.000 | 0.000 | 0.000 |
| 20000 | 0.000 | 0.000 | 0.000 |
| 40000 | 0.000 | 0.010 | 0.000 |
| 80000 | 0.016 | 0.020 | 0.010 |
| 160000 | 0.040 | 0.050 | 0.028 |

## Quicksort using Random Pivot

|  |  |  |  |
| --- | --- | --- | --- |
| Items | Ascending | Random | Descending |
| 100 | 0.000 | 0.000 | 0.000 |
| 1000 | 0.000 | 0.000 | 0.000 |
| 10000 | 0.132 | 0.000 | 0.140 |
| 20000 | 0.540 | 0.000 | 0.560 |
| 40000 | 2.182 | 0.010 | 2.250 |
| 80000 | 8.742 | 0.020 | 9.008 |
| 160000 | 34.958 | 0.040 | 36.058 |

# Observations

Quicksort using a naive pivot ran worse on ordered data than on unordered data. Datasets ordered in ascending and descending order were the slowest, thus being the worst case. Randomly ordered data, on the other hand, had the best case. The best case, average case and worst case are O(n^2), O(n log n) and O(n^2), respectively.

Quicksort using a random pivot behaved similarly to that of using the naïve pivot. There was a small improvement (1 microsecond) improvement with the ascending datasets, but it wasn’t enough to categorise it into another complexity class.

Quicksort using a median of three pivot saw a huge improvement in all cases, compared to other two algorithms. I observed a time complexity of O(n log n) for ordered datasets and O(log n) for randomly ordered datasets.