Lab 12 Report

Date: 4/27/20

Group: Wednesday Group 08

Group Members: Alan Joshua, Isaac Krisch, Zaina Qasim

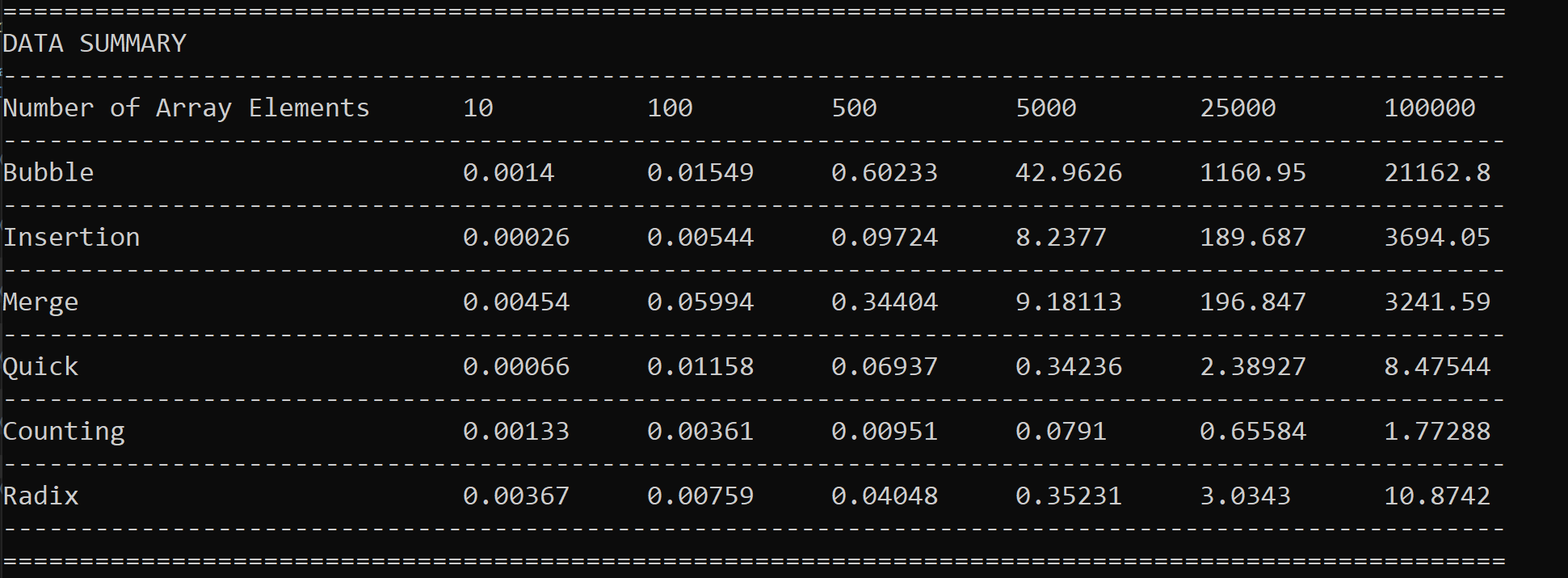
# Objectives/Concepts explored and their Importance in Computer Science

The objective of this lab was to investigate the actual performance of different sorting algorithms. We explored six different types of sorting algorithms, namely, Bubble Sort, Insertion Sort, Merge Sort, Quick Sort, Counting Sort, and Radix-Sort and analyzed their time and space complexities.

# Task 2: Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sorting algorithm/array size | Time (in ms) for 10 runs | | | | | |
| 10 | 100 | 500 | 5000 | 25000 | 100000 |
| Bubble sort | 0.0014 | 0.01549 | 0.60233 | 42.9626 | 1160.95 | 21162.8 |
| Insertion sort | 0.00026 | 0.00544 | 0.09724 | 8.2377 | 189.687 | 3694.05 |
| Merge sort | 0.00454 | 0.05994 | 0.34404 | 9.18113 | 196.847 | 3241.59 |
| Quicksort | 0.00066 | 0.01158 | 0.06937 | 0.34236 | 2.38927 | 8.47544 |
| Counting sort | 0.00133 | 0.00361 | 0.00951 | 0.0791 | 0.65584 | 1.77288 |
| Radix-sort | 0.00367 | 0.00759 | 0.04048 | 0.35231 | 3.0343 | 10.8742 |

### Screenshot:

 This TEST WAS RUn IN RElease MODE (RUNTIMES MAy be LONGER IN DEBUG MODE)

# Task 2: Graph

# Task 2: Discussion

Using the table and the graphs above, we can infer that Radix Sort, Counting Sort and Quick Sort work much faster than the remaining three. As expect both bubble sort and insertion sort took the longest amount of time, give their average time-complexity of O(n^2). Merge sort, on the other hand has an average run-time complexity of O(nlog(n)), and it performed slightly better than Insertion Sort. Quicksort too has an average runtime complexity of O(nlog(n)), however it performed much better than Insertion sort. This can be attributable to better space complexity or the algorithm itself which seems to be working much better on the test data. Given different data, the test might yield different results. Finally. Counting sort and radix-sort had the best performances. Given their runtime complexities of O(nk) and O(n+k), respectively, it was expected that the two algorithms would be more efficient.

# User Instructions

It its advisable that this program should be run in release mode to minimize runtime.

# Contributions

All members contributed an equal amount.