Muhammad Ashraf

PA03-MysterySorter Write-up

A group of troublesome hackers wronged Dr. Fontenot. Our beloved professor was working tirelessly on a collection of sorting algorithms and those hooligans changed the names of the functions. Dr. Fontenot specifically entrusted me to fix this problem. In PA03, I tested each of the mystery sorts against a collection of datasets with specific sizes. I had the datasets in sorted, reversed, and random orders. I utilized the std::chrono library to map the times of the algorithms after they were implemented. I used the times and plotted them with their respective algorithms and tested them against the behaviors of the five algorithms in order to reveal the identities of the algorithms. I created three graphs in a spreadsheet, each of the graphs displayed the times for sorted, reverse, and random orders based on the mystery sorts.

For *mystery01*, the times displayed were the fastest compared to all of the other sorts. The line plotted on all three of the graphs for *mystery01* was exhibiting a nlogn behavior. The line was also closest to the x-axis which just shows how fast this algorithm performed. With this information, I was able to deduce that *mystery01* is Quick sort.

For *mystery02*, the behavior displayed on all three graphs was shown to do the worst under all three sort orders and it performed the slowest compared to all of the other algorithms. It was exhibiting an n^2 behavior. With this information, I came to the conclusion that *mystery02* is Selection sort.

For *mystery03*, the algorithm performed well with the sorted and random orders. However, when it was tested against the reversed order, the algorithm did not perform well. This algorithm showed n^2 behavior when the dataset was reversed and random. I concluded that *mystery03* is Insertion sort. I'm confident in this because the algorithm performed very well with the sorted dataset. Insertion sort does not do well when a dataset is in reverse order.

For *mystery04*, the algorithm exhibited nlogn behavior for the sorted, reverse, and random orders. During my testing, I noticed that this algorithm did not do well with large datasets. The program would break if I used very large datasets in the millions. Merge sort does not do well with larger datasets and so that led me to conclude that *mystery04* is Merge sort

For *mystery05*, the algorithm performed well with the sorted order. However, with the reverse and random order, we can observe that the algorithm is showing n^2 behavior. *Mystery05* is Bubble sort because it performs well with a sorted set, but when the set is reversed or random it goes into n^2 behavior.

By testing the five mystery sorts with the std::chrono library and plotting the times, I was able to identify what each of the sorts were.