

Assignment 6: Sorting

Nayeli Castro
December 14, 2020

The algorithms used in this assignment were quick sort, merge sort, selection sort, insertion sort, and bubble sort. Quick sort and merge sort have the best runtime in the best scenario out of the 5 algorithms with a runtime of $O(n \log n)$. Selection sort and insertion sort run at a runtime of $O(n^2)$ in the average case, but may be improved to run more efficiently. Bubble sort is the worst sorting algorithm out of the set with a runtime of $O(n^2)$ in every case. Bubble sort is typically the slowest because of this.

I selected a file with doubles and ran the program I made which consists of these 5 algorithms. Merge sort had the duration time of 19.771, making it the fastest of the algorithms in my program. Quick sort had the duration time of 21.63, which is still fast, but not as quick as merge sort. Insertionsort had the duration time of 6723.06. There was a big jump in time between insertion sort and selection sort, as selection sort had a duration time of 11398.6. The slowest algorithm was predictably bubble sort with a time of 51000.8.

There are several tradeoffs that are involved when picking a sorting algorithm to use. Typically, people would lean towards using the fastest sorting algorithm due to it being immediately appealing. However, despite merge sort's speed, it is complex to write and consumes large amounts of memory, as shown through the segmentation faults I encountered while writing the code. Quick sort is a simpler algorithm to write than merge sort. Insertion and selection sorts are even simpler to write than quick and merge sort, and despite their less appealing runtime, their flexibility makes up for it. Bubble sort is the easiest algorithm to write, however it is slow.

I believe C++ was a beneficial choice of language for these algorithms. As opposed to languages such as Java or Python, you have control of memory allocation in C++ and makes writing these algorithms more flexible.

The data of these program runs are not completely accurate due to its dependency on the user's computer. It is more efficient to calculate the efficiency of the programs through mathematical means.