Raymond Yan
CPSC-350
Prof German
12/10/2020

**Experiences on Implementing and Testing Different Sorting Algorithms**

The time difference between different sorting algorithms is far more drastic than I expected. With a relatively smaller data size, for example, 50 elements, the differences are not observable if no timestamping with precision to milliseconds has been used. However, with a much larger data size with more than 10000 elements, the difference in operating time is so obvious among the 5 algorithms that we don't even need a timestamp to help us to observe it: as the data size become bigger and bigger, the "simpler" algorithms takes longer and longer and longer and longer (like around 10, 15 or even 30 seconds for 10000 elements) time than the more complicated algorithms (takes less than 1 second for the same amount of data). The difference between O(nlogn) and O(n^2) is significant.

In picking one algorithm over another, we have to consider many factors: time complexity, space complexity, amount of code, difficulty of implementation, etc. If we choose "simpler" algorithms with smaller amount of code and easier code and procedures, the tradeoff is huge amount of time if we need to sort huge amount of data; In the other hand, if we only need to sort 10 numbers but implemented something like a Merge Sort of a Quick Sort, it doesn't worth all the work we do to code them.

I only used C++ for coding the sorting algorithm now, but I'm sure if I could use another more complexed language like JS or python, the process would be much easier since I can then use much developed data structure to assist my codes and due with less primitive level problems directly like figuring out the addresses and pointers.

A shortcoming of this empirical analysis is that it takes a huge amount of time for me to set everything up until I can finally perform the test. The result is approximately the same to the big-Oh theoretical analysis, yet the big-Oh can be calculated in seconds.