

Michael Efseaff

Assignment 7 write up

The time differences were way more dramatic than I thought. Thinking about something being $O(n^2)$ vs $O(n \log n)$ in my head made me think that the difference wasn't actually that large since I can't imagine in my head the functions growing over a few hundred thousand points. Especially when it came to bubble sort which for 500,000 doubles, took about 7-8 minutes on my computer, while it was just a fraction of a second with quick sort, merge sort, and shell sort. I think the major trade offs when it comes to picking one of the sorting methods over another is mainly, CPU usage, Memory usage, and difficulty to implement. For CPU usage the 3 slower algorithms (bubble, insertion, and selection) , according to docker, used 100% of the CPU for the entire duration of the sorting algorithm. While the 3 faster algorithms (merge, quick, and shell) didn't seem to peak the CPU usage over 60%. For memory usage merge sort is the obvious outlier when it comes to eating up memory since it makes so many arrays. However for the purposes of this assignment I didn't notice any spike in memory usage during run time, however this could just be because the data sets aren't in the millions or billions. For complexity the three slower algorithms were much more simple and straightforward to write, especially bubble sort, while merge sort and quicksort really took some thinking and research to understand what is supposed to be happening in the code. I think C++ was more of a detriment for this assignment because I had to manually create a new array for each sorting algorithm, which takes the program longer than some of the actual sorting algorithms themselves, while other languages might have built in methods for arrays to copy over data. The shortcomings of this empirical analysis is that

everyone's machines are different and as such the algorithms will run differently on each person's computer and docker container, additionally we are limited by the data set and can only put in so many doubles in the arrays to test the sorting algorithms, which might not show their long term efficiency accurately.