

### Assignment 6 Analysis

The time differences were definitely more drastic than I expected. I wasn't expecting for merge sort and quick sort to be sorting thousands of numbers in a hundredth-thousandth of a second (i.e. 0.000956 seconds for 10000 numbers), while insertion, bubble, and selection sort were sorting thousands of numbers closer to one-tenth of a second (i.e. 0.042483 seconds for 10000 numbers).

If we were to choose Merge Sort, the tradeoff for speed would be memory because it requires an additional array to store temporary data that is being sorted. If we were to choose Quick Sort, the tradeoff would be for its worst-case performance of  $O(N^2)$ . It doesn't require additional memory like Merge Sort, but there is a chance that we will run into slow performance for a large set of data. For Insertion Sort, the downfall is that it has a slow average runtime of  $O(N^2)$ . However, for nearly sorted data, the runtime is  $O(N)$  and is even faster than Merge Sort. For Selection Sort, there aren't many advantages to it besides the fact that it is an "in-place" sorting algorithm; therefore, it doesn't require additional memory. The tradeoff is its performance of  $O(N^2)$  runtime. For Bubble Sort, the primary advantage of this algorithm is that it's easy to implement, so you can take advantage of sorting in a scenario where you are in a remote area without notes. Its tradeoff is its performance. Bubble sort has a runtime of  $O(N^2)$  and is very slow for lists with huge numbers of items.

I would say that choosing to sort in C++ is faster than other programming languages because we have control over memory allocation. Thus, this allows C++ programmers to be more efficient when sorting because we are being meticulous with passing references to arrays or pointers to arrays. Some shortcomings to this empirical analysis is the efficiency of it. We have to test multiple data sets with varying inputs before we can conclude our findings. Also, if we have an immensely large set of data, sorting would take up lots of memory, CPU, and time. Empirical analysis is also dependent on many variables, so depending on our RAM, results may differ when sorting.