The most prominent thing I noticed was that when the inputs were already partially sorted , it would drastically change the runtime comparison. The more unsorted the numbers, the less efficient insertion sort was. In bubble sort, the presorted numbers were far more efficient at sorting themselves than the values that were not presorted. Selection sort was the outlier for this in that it seemed to have an even output no matter how sorted the values were.

Quick sort was the most consistent algorithm and is thus, now my new favorite. I found the bubble sort to be the easiest to implement and that it worked better with smaller random sequences than larger presorted sequences. Insertion sort worked similarly and was also not too difficult to implement.

All in all, I think that Quicksort is the most optimized algorithm. Its recursive qualities enable it to be super competitive with other algorithms in all case scenarios and to be unrivaled for large input problems. Also, quick sorts worst case scenario runtime remains competitive with other algorithms best case scenarios, making it easily the best algorithm

It seems as the the input size, whether or not the sequence is presorted or not already, and the memory available are the three criteria one should consider when deciding which algorithm is best. For most situations, the quicksort is the best..