Throughout this project there was a common factor in the way that the in place algorithms worked. All of the algorithms had a wall, pivot, and comparator. With this concept they were different in the way of optimizing it from the bubble sort that had nothing set in stone until it finished all of it and was done switching making it unpredictably long at higher list. On the other side the selection went through and took the same amount of speed through any amount just taking longer on the amount of comparators making it at times taking a while. Then there was the insertion sort that was a modified version of the selection choosing the one but instead placing it making it faster in most cases than the selection slightly. Then we have the quick sort that took a more dynamic and recursive standpoint to take the array into smaller parts and compare cutting the speed in large cases. I love the shell sort because of the way it goes through and sorts and is a playoff of the selection and insertion and close to the speed of quicksort in most cases. Then we get to the Tim sort which is brute force and the fastest sort but 700 lines of code and very stable.