In sorting, I learn the bubble sorting is the slowest and inefficient code. Creating a new variable and not to mention swapping up and down the array one by one over and over in the loop. Having N^2 making it the slowest out of all the sorters. In reverse order I’ve expect to 10,000,000,000 we got a nice close result of 6,749,270,067. Another example of bubble sorter running at its best. I would expect 100,000 but gotten 75,553 close to its N number of integer.

Moving to the second sorter, “Selection Sort” this program gave me interesting results in its worst possible. It did well with 2,319,831,276 making it better than bubble sort. Due to the face in the program it looks for the lowest value in its array then but it the back. Slowly cutting down search one by one of its N times it ran. The bubble sort beat selection sort in its best possible condition. I believe due to the face the selection sorter will run over and over looking for the lowest value to put behind. This will cause the program to lose to bubble sort. If there are no swap, then bubble sorter ends.

Going to the finally sorter, “Insertion Sort.” In its worse condition the program is slower than selection sort. Having the time 3,054,261,493 could be the fact that the program uses selected number and compare to its array from highest to lowest and insert it in. Insertion sort with 1,450,378,972 vs Bubble sort with 11,960,441,794 vs Selection Sort with 2,207,980,578. These are with random array. Insertion sort is the best out of all of them. Could be that Insertion using its rearranging the array as it goes. Instead of going through the for loop many times. Looking for the lowers number like Selection or Bubble with it switch as it goes up until there are no swaps.