Conjecture:

I believe that after studying bubble sort, insertion sort, and selection sort. I believe that the best will be insertion sort since the code appears to stop doing compares for numbers if the code notices that the number its comparing to is less than the index. Due to being able to exit out from the code more quickly. I believe that the insertion sort will be the quickest. Also selection sort might be also slower due to possibly having to compare each number to all the other numbers without exit.

Data after testing

Bubble Sort:

10 Accounts: 3,400 ns

100 Accounts: 226,800 ns

1,000 Accounts: 6,756,800 ns

100,000 Accounts: 15,286,776,799 ns

1,000,000 Accounts: Unable to Finish (threshold)

10,000,000 Accounts: Unable to Finish

Selection Sort:

10 Accounts: 2,800 ns

100 Accounts: 124,601 ns

1,000 Accounts: 5,169,500 ns

100,000 Accounts: 2,235,530,001 ns

1,000,000 Accounts: 226,494,987,800 ns

10,000,000 Accounts: Unable to Finish (threshold)

Insertion Sort:

10 Accounts: 1,900 ns

100 Accounts: 103,799 ns

1,000 Accounts: 4,490,499 ns

100,000 Accounts: 2,983,726,300 ns

1,000,000 Accounts: 307,803,137,500 ns

10,000,000 Accounts: Unable to Finish (threshold)

Specs:

CPU: Ryzen 5 1600

Ram: 16gb

Conclusion:

My conclusion turns out to be slightly correct and slightly wrong. It seems that insertion sort was faster than both selection sort and bubble. However when the data array gets very large, the selection sort seem to out perform the other two sorting algorithms. I believe this occurs due to how many times the data need to be moved around during sorting and whether the array is nearly sorted. If the array is nearly sorted, then the insertion sort will not have to move data down the array that many times and will exit out of the loops early. However in a bigger arrays, the data will tend to have a higher chance of not being nearly sorted which will cause the data needing to be moved all the way to start of an array if a small value is found at the end of array. This can impact the performance of the sorting algorithm and cause a slower response time.