

Exercise 2.1.7: Which method runs faster for an array in reverse order, selection sort or insertion sort?

Solution: Reverse order is the absolute worst case for insertion sort as it'll have to do $\frac{n*(n-1)}{2}$ exchanges as it checks each index and swaps with everything previous. This is because previous elements are all larger since the array being scanned is given in reverse or descending order. Best case for insertion sort occurs when the array is already sorted as it then does n -comparisons but no swaps since already in order.

Selection sort does not care about the order the array is in. It will do $N-1$ changes since it finds the smallest value moving forwards. So regardless, it will do $N-1$ changes. Although they both have $O(N^2)$ complexity in worst case, selection sort would still be better for this problem because it does less swaps compared to insertion sort.