Sort

Insertion sort.

3. Loops through the array starting from index 1. Inserts the value into the left side of the array, which is always ordered.

5. The degree of unsortedness affects time of execution directly - in best case inner loop never done.

Selection sort.

3. Loops through the array from index 0, finds minimal value and swaps it with index 0. Repeat starting from index 1 and so on.

5. Does not depend on the input array sortedness.

Bubble sort.

3. Pushes values to the right until the next element is greater than the value.

5. It has a stop condition, so it ends as soon as the tail of the array is sorted. In the worst case – slow.