Practical 4

Q1.How many compares does insertion sort make on an input array that is *already sorted*?

Constant	
Logarithmic	
Linear	X
Quadratic	

Q2. What is a stable sorting algorithm?

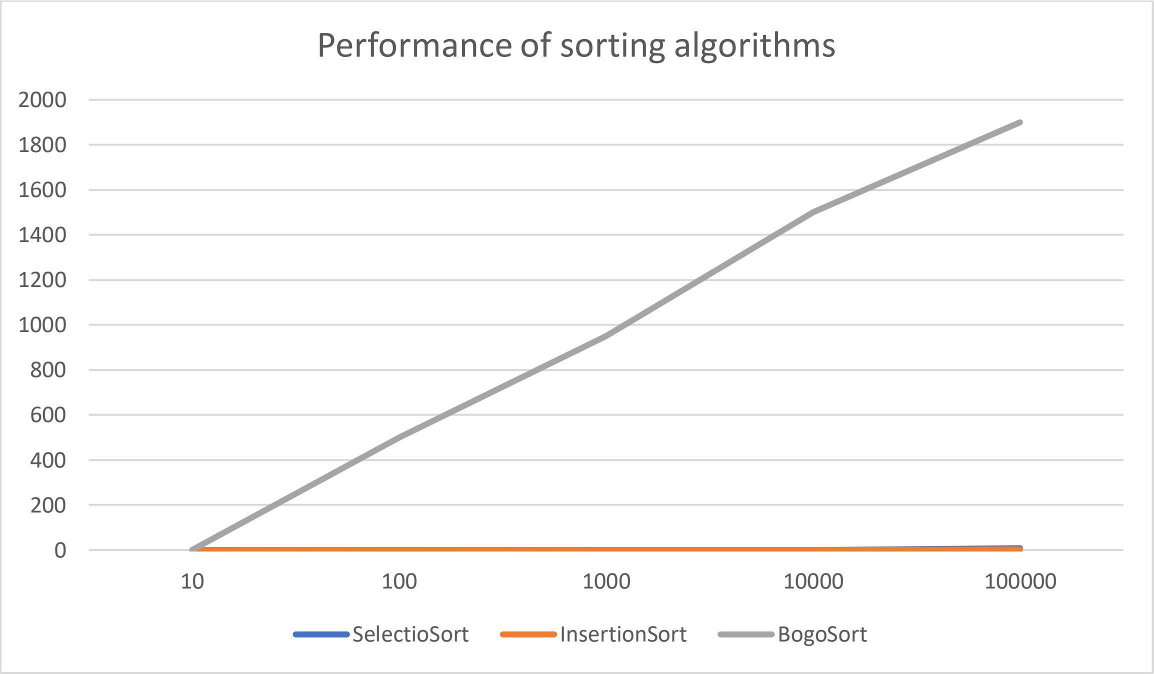
It means that the relative order of the elements is maintained with sort keys.

Q3. What is an external sorting algorithm?

A. Algorithm that uses tape or disk during the sort

Q4. Identify 6 ways of characterizing sorting algorithms?

1. By time complexity
2. Space complexity
3. Internal or external algorithm
4. Swapping or inserting
5. Recursion or non-recursion
6. Stability



- Selection sort has a time complexity of O(n^2) as it contains two nested for loops. This isn't the most time efficient sorting algorithm as seen on the graph, when input size increases the time it takes to complete the sort increases rapidly.

- Insertion sort has a time complexity of O(n^2) as it contains a for loop and a nested while loop. As the selection sort the algorithm isn't very efficient but in my testing it showed that insertion sort is slightly more efficient.

- Silly sort has a time complexity of O(∞) as it uses random sorting to sort the array. There is no upper bound on this algorithm and in theory it could run forever. This is by far the worst algorithm to use if you want to sort you array in a timely manner.(The timing graph for bogoSort is an approximation as it would take too much time to compute)