Lab 4. Traditional Sorting Algorithms

1. Implement the following traditional sorting algorithm in Java and test your code using the following array *testA*, *testB*.

1.1.Selection Sort

1.2.Rank Sort

1.3.Insertion Sort

.testA = [1, 62, 81, 0, 23, 55, 76, 87, 20, 54, 65, 76, 1]

.testB = [(71, 2); (64, 8); (31, 56); (98, 1); (3, 6); (59, 837); (49, 58); (61, 8)], where we consider the first number is the key element

2. Now we are going to look at a new comparison sorting algorithm, the idea behind this algorithm is:

1. It is a simple sorting algorithm.
2. It will repeatedly step by step go through the whole list, comparing each pair of adjacent items and swaps them if they are in the wrong order.
3. The process of going through the list is repeated until no swaps are needed, which indicates that the list is sorted.

Try to understand the idea and implement it in Java. Use *testA* and *testB* to validate your program. What is worst and best complexity for your program in big O notation?