

Series of Exercises 03

Exercise 01:

The pseudocode of a sorting algorithm is as follows:

```
Arraysort1 (data [], n)
  for i = 1 to n-1
    Move all elements data [j] greater
    than data [i] by one position;
    Place data [i] in its proper position;
```

1. What do you guess about this sorting algorithm (its name)?
2. Propose a correct implementation of this algorithm.
3. What do you guess about the time complexity of this algorithm in the best and worst cases?

Exercise 02:

The pseudocode of a sorting algorithm is as follows:

```
Arraysort2 (data [], n)
  for i = 0 to n-2
    Select the smallest element among
    data [i], . . . , data [n-1];
    Swap it with data [i];
```

- The aforementioned questions of the Exercise 01 apply to this pseudocode.

Lab session: Try to implement in C++ all of the sorting algorithms that have been listed in the tutorial session.

Exercise 03:

The pseudocode of a sorting algorithm is as follows:

```
Arraysort3 (data [], n)
  for i = 0 to n-2
    for j = n-1 down to i+1
      Swap elements in positions j and
      j-1 if they are out of order;
```

- The aforementioned questions of the Exercise 02 apply to this pseudocode.
5. What do you think about the disadvantages of this algorithm?
 6. Propose an optimized version to overcome that disadvantage.

Exercise 04

Write a code fragment that sorts the elements of an array in ascending order using the merge sort algorithm.

Exercise 05

Apply quick sort to partition (mention all the steps) and sort the following array:

8	5	4	7	6	1	6	3	8	12	10
---	---	---	---	---	---	---	---	---	----	----