

## LAB GUIDE. SESSION 2

---

### GOALS:

- **Sorting algorithms and their comparative study**

### 1. Three bad sorting algorithms

In the Java files associated with this lab you have three sorting algorithms already studied in class (**insertion**, **direct selection** and **bubble**). **The specific code of the three methods should be included in the files by the student.**

There are bad algorithms because there are quadratic –  $O(n^2)$  – in their best, worst and average case (except the insertion algorithm, which in the best case is linear  $O(n)$ ).

To prove that all methods work correctly, a `SortingTests` class is provided. It has an argument `n` that is the size of the problem. Try to understand in detail the operation of all the algorithms by analyzing the times for different sizes of the problem.

A class `SortingMeasurements` is also provided. You should parameterize it to correctly measure the respective times in the different cases.

### 2. A better sorting algorithm: Quicksort

In this case, you are going to study the **Quicksort** sorting algorithm. You should study it in detail, as it is a much more elaborate algorithm than the others. **Complete the code** when necessary and analyze the times for different sizes of the problem. Finally, conclude whether the times obtained are the expected from the complexity in each case.

- `QuicksortMedianOfThree.java` → It has an argument `n`, that is the size of the problem. It is the version we saw in class.
- `QuicksortFateful.java` → It has an argument `n`, that is the size of the problem (it uses a bad pivot). It is usually a very bad choice.
- `QuicksortCentralElement.java` → It has an argument `n`, that is the size of the problem. In this case, instead of the median of three we use as the pivot just the central element.

### TO DO:

#### A. Work to be done

- An `algstudent.s2` **package** in your course project. The content of the package should be:

- All the files that were given with the instructions for this session but completing the fragments that were incomplete in `Bubble.java`, `Selection.java`, `Insertion.java` and `QuicksortCentralElement.java`.
- A **PDF document** using the course template. The activities of the document should be the following:

- **Activity 1. Time measurements for sorting algorithms.**

- Four tables with times for each of the algorithms (Insertion, Selection, Bubble and Quicksort with the central element as the pivot). An example of one of the tables is below:

| $n$                                 | $sorted(t)$ | $inverse(t)$ | $random(t)$ |
|-------------------------------------|-------------|--------------|-------------|
| 10000                               | .....       | .....        | .....       |
| 20000                               | .....       | .....        | .....       |
| 40000                               | .....       | .....        | .....       |
| 80000                               | .....       | .....        | .....       |
| 160000                              | .....       | .....        | .....       |
| 320000                              | .....       | .....        | .....       |
| 640000                              | .....       | .....        | .....       |
| 1280000                             | .....       | .....        | .....       |
| ...                                 | .....       | .....        | .....       |
| <i>Until an exception is thrown</i> |             |              |             |

- A brief explanation (a paragraph) for each of the tables to conclude whether the values make sense regarding the expected theoretical complexity.
- **Activity 2. QuicksortFateful.**
  - Briefly explain what the criteria is for selecting the pivot in that class. Indicate when that idea can work and when that idea will not work.

## B. Delivery method

You should include in your Java project a new `algstudent.s2` package with the following content inside it:

- All the requested source files.
- The requested PDF document called `session2.pdf` with the corresponding activities.

**Deadlines:**

- The deadline is one day before the next lab session of your group.