ALGORITHMICS - Bachelor of Software Engineering, 2025

LAB GUIDE. SESSION 5

GOALS:

- **Problem solving with dynamic programming**

1. The minimum path

Many real problems can be modeled through a graph; in this case, we will focus on a graph with the following characteristics:

- It is a directed graph.
- There are no loops, that is, edges with the same origin as destination.
- Between any two different nodes there is a probability **p1** that an edge exists and a probability **p2** (p2=1-p1) that it does not exist.
- In the case that an edge exists, the weight is positive and has a value in the interval **[minWeight … maxWeight]**.

YOU ARE REQUESTED TO:

Efficiently solve the calculation of the **shortest path between any pair of different nodes** (source <> destination).

To do this, you must **implement**:

PART A:

A class **MinimumPathsExample.java**, that will calculate and write the minimum costs for a simple graph of 5 nodes (see code provided that can be completed and changed as you wish). The objective of this class is to check (when programmed) whether the written paths are correct or not.

When executed, it must show something like the following:

```
FROM NODO0 TO NODO1 = NODO0-->NODO1
MINIMUM COST=19
**************
FROM NODO0 TO NODO2 = NODO0-->NODO2
MINIMUM COST=10
**************
FROM NODO0 TO NODO3 = NODO0-->NODO2-->NODO3
MINIMUM COST=24
```

```
**************
FROM NODO0 TO NODO4 = THERE IS NO PATH
**************
FROM NODO1 TO NODO0 = NODO1-->NODO2-->NODO3-->NODO0
MINIMUM COST=61
**************
FROM NODO1 TO NODO2 = NODO1-->NODO2
MINIMUM COST=20
**************
FROM NODO1 TO NODO3 = NODO1-->NODO2-->NODO3
MINIMUM COST=34
**************
FROM NODO1 TO NODO4 = THERE IS NO PATH
**************
FROM NODO2 TO NODO0 = NODO2-->NODO3-->NODO0
MINIMUM COST=41
**************
FROM NODO2 TO NODO1 = NODO2-->NODO1
MINIMUM COST=19
**************
FROM NODO2 TO NODO3 = NODO2-->NODO3
MINIMUM COST=14
**************
FROM NODO2 TO NODO4 = THERE IS NO PATH
**************
FROM NODO3 TO NODO0 = NODO3-->NODO0
MINIMUM COST=27
**************
FROM NODO3 TO NODO1 = NODO3-->NODO2-->NODO1
MINIMUM COST=40
**************
FROM NODO3 TO NODO2 = NODO3-->NODO2
MINIMUM COST=21
**************
FROM NODO3 TO NODO4 = THERE IS NO PATH
**************
FROM NODO4 TO NODO0 = NODO4-->NODO1-->NODO2-->NODO3-->NODO0
MINIMUM COST=141
**************
FROM NODO4 TO NODO1 = NODO4-->NODO1
MINIMUM COST=80
**************
FROM NODO4 TO NODO2 = NODO4-->NODO1-->NODO2
MINIMUM COST=100
**************
FROM NODO4 TO NODO3 = NODO4-->NODO1-->NODO2-->NODO3
MINIMUM COST=114
**************
```

PART B:

A class **MinimumPaths.java**, that will calculate and write the shortest paths for a graph with **nodes** whose matrix of edge weights is randomly generated, with these values of the fundamental parameters mentioned above:

p1=0.5; p2=0.5; minWeight=10; maxWeight=99

PART C:

A class **MinimumPathsTimes.java**, that will grow in graph size like this: **n=200, 400, 800, …,** until it takes more than 5 minutes.

The weight matrix, for each **n**, is randomly generated according to the previous section.

The timing for each size **n** will encompass all operations: randomly generating the matrix, computing Floyd, and computing the shortest paths for every pair of different nodes.

The times will be measured WITHOUT OPTIMIZATION (with **-Xint**) and to avoid unwanted writing in this case, the writing statements (**System.out….**) that were used in the previous section to write the minimum paths, will be commented.

PART D:

Finally, place the times obtained in the previous section into a table and explain whether they correspond (or not) to the complexity attributed to the algorithm.

2. Work to be done

- An `algstudent.s5` **package** in your course project. The content of the package should be the Java files used and created during this session.
- A `session5.pdf` **document** using the course template (the document should be included in the same package as the code files). You should create one activity each time you find a "YOU ARE REQUESTED TO" instruction.

Deadline: The delivery of this practice will be carried out, in time and form, according to the instructions given by the lab teacher.