

Project

Graph and shortest paths

Instructions:

- This project has to be made by groups of 3 students. There is no possibility to work alone
- Prepare a ZIP archive containing your source code in a “src” folder, a written report in a pdf file and a ppt file and upload it to moodle. Name of the ZIP: member1-member2-member3.

Objectives

- The main objective is to use the algorithmic and programming tools treated in the lectures and tutorial courses in a real graph and to use generics and collections.

A. Dijkstra algorithm for weighted digraphs

Use the previous TP project based on V and E for nodes and edges. Add a class called *DijkstraSP*. This class will implement the Dijkstra algorithm for detecting shortest paths in weighted-digraphs. This class will contain the following functions:

1. 3 arrays: `boolean[] marked`, `int[] previous` and `int[] distance`
2. A function called `verifyNonNegative(WDGraph G)` which takes as input a weighted-directed graph and verifies that all weights in the graph are non negative.
3. Create a function called `DijkstraSP(WDgraph G, int s)` which implements the Dijkstra algorithm for shortest paths studied in the lecture. The input arguments are a weighted-digraph and a root vertex `s`.
4. As for the previous section, create the functions `hasPathTo(int v)`, `distTo(int v)` and `printSP(int v)`.

Test the previous functions with the graph *graph-WDG.txt*.

B. Improve Dijkstra algorithm for weighted digraphs

- Try to implement 2 improvements of Dijkstra or its alternatives and compare their execution time and complexity with normal Dijkstra. Use the metro map of Paris in moodle to compare the algorithms.
- In the first line, the first number shows the number of nodes, and the second number indicates

the number of edges.

- After the first line (but before the separator), each line indicates a node (station) in the network, and its unique number. Format: 0009 Assemblée Nationale
- There is a separator line with only a \$ character to separate the previous bullet point from the next one.
- After the separator, each line indicates a tunnel between two stations (an edge) and the distance of that tunnel. Format: 9 78 62 (meaning that there is an edge connecting node 9 and node 78, and it has a distance of 62).