Algorithms & Data Structures
SI3 - Polytech Nice-Sophia - Edition 2018

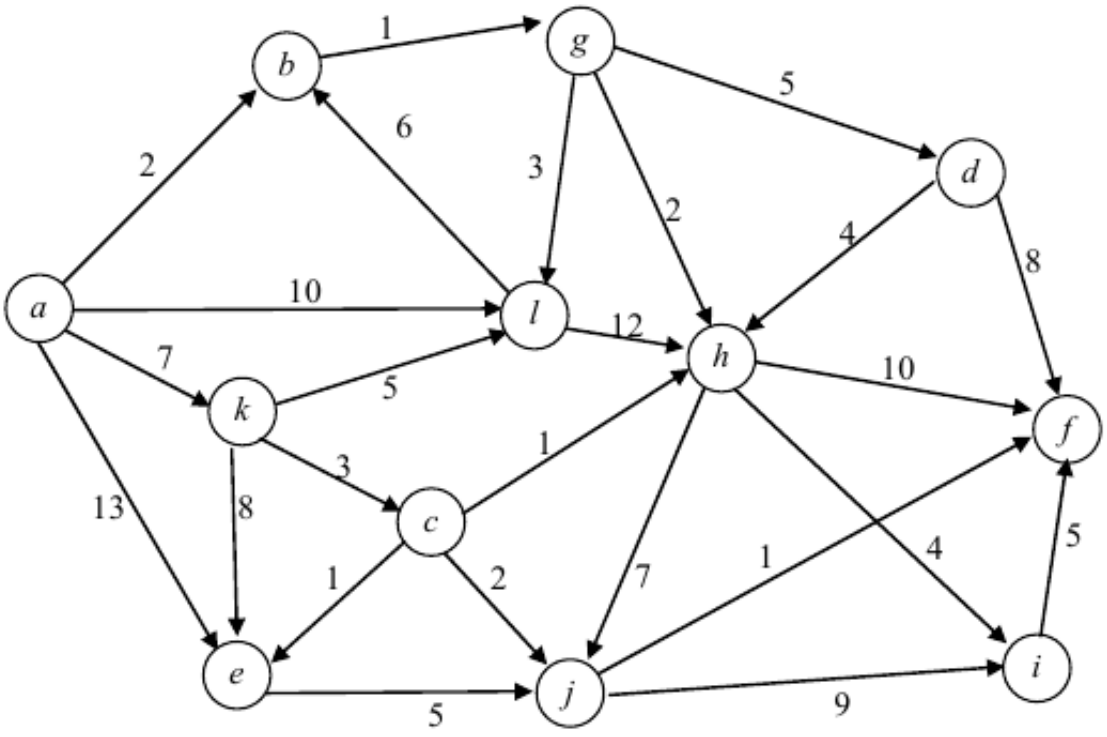
Lab #10: Shortest Paths

This lab will give you practice about directed weighted graphs and Dijkstra shortest path algorithms. For all parts you are to use the provided material:

- graph.py: this is an updated version of the `Graph` class we've been using. This new version just have the missing function `setValue` on `Vertex` object
- BinaryHeapSpecial.py: this is the new binary heap class supporting the `decreaseKey` method. You are to complete this class
- lab10.py: you are to write the functions `dijkstra` and `print_paths` from this file.

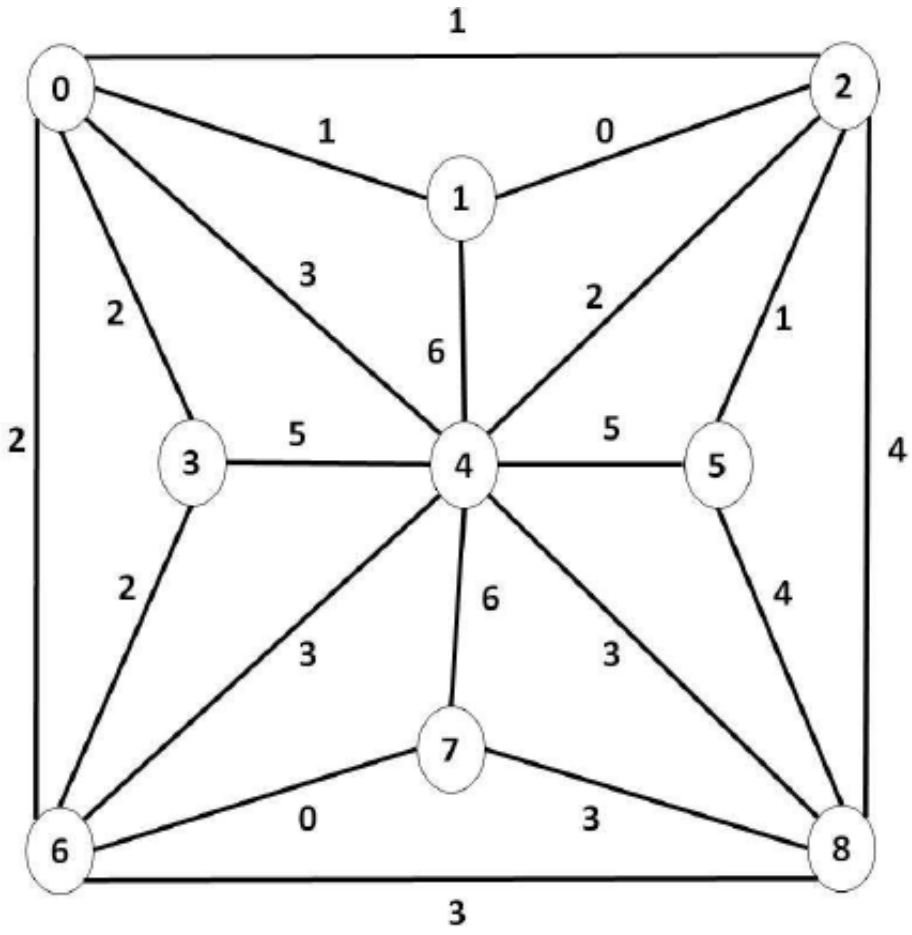
Part 1

Run the Dijkstra algorithm on the following graph starting from vertex **a** and show the result. Show the table as the one shown in class after 4 vertices are known



Part 2

Run the Dijkstra algorithm on the following graph starting from vertex **0** and show the result. Show the table as the one shown in class after 6 vertices are known



Part 3

Given a directed positive weighted graph **G** and a vertex **v** of **G**, give an efficient algorithm to compute the shortest paths from **all vertices** to **v** (just explain the algorithm in English, don't write any code).

Part 4

In this part, you are to implement the Dijkstra algorithm. To implement the Dijkstra algorithm efficiently you need to use a special heap class which supports the `decreaseKey` operation. This method allows to change the cost of a vertex which is already inside the heap. Check carefully the `BinaryHeapSpecial` class template provided and try to complete it before you write the function `dijkstra`.