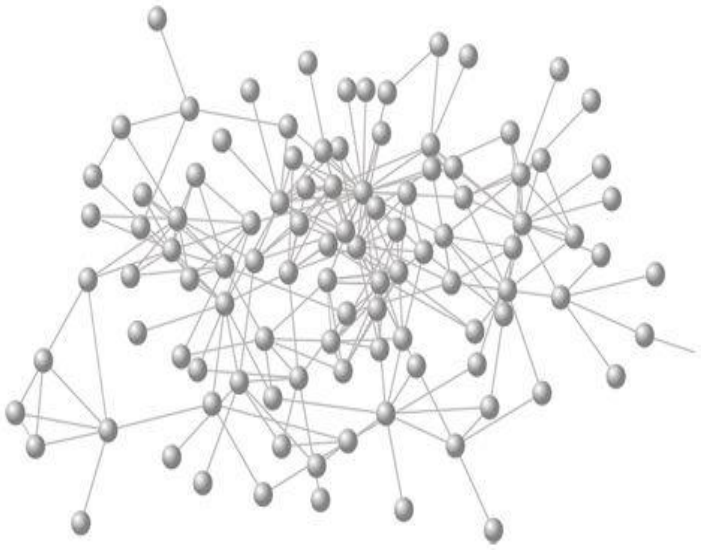
CSCE 2211 Fall 2023 Applied Data Structures
Assignment #6

Dr. Amr Goneid ***Date: Sat Dec 2, Due: Tue Dec 12, 2023***

This is a Bonus (Optional) assignment. It will count as 5% of the total course grade.
No submissions will be accepted after the due date.

The problem: Finding all shortest paths in a weighted graph

A network of highways connects N cities. Such network can be represented by a graph with N nodes and the weights on the edges represent the distances between pairs of cities. An Excel sheet **"CitiesG.xlsx"** gives the adjacency matrix representation for such weighted graph for N = 14 cities. The graph is connected, and the weights are all positive integers in the range 20 – 100. Zero weights represent the absence of a highway, or the distance between a city and itself. The cities are simply named (A, B, C, ….).



Develop a program to:

1. Traverse the given graph using the ***DFS*** algorithm.
2. Determine the shortest paths from a given city (e.g., A) to all the other cities in the given graph using **Dijkstra's** algorithm.

Notes on the design and implementation

- Use the adjacency matrix representation of the graph, i.e., a 2-D array of size V_{max} by V_{max} where V_{max} is the maximum number of vertices (e.g $V_{max} = 50$).
- The vertices are given numbers (0,1,….V-1) where V is the actual number of vertices in the graph. These numbers can be mapped to names (e.g. A , B , C…).
- An edge (u,v,w) has one vertex u, a second vertex v and a weight w (a positive integer). It is represented as an object of a class "Edge". This class will also contain the definition of weight type (integer). A 1-D array of size E_{max} is used to store the

non-zero edges in the graph, where E_{max} is the maximum number of possible edges = V_{max} *(V_{max} -1)/2. The actual number of such edges in the graph will be E so that the edges will be stored in locations (0 .. E-1).

- A class "***Graphs***" represents the above ADT. The "skeletal" class header and implementation files are given in files **"Graphs.h"** and **"Graphs.cpp"** in the zip file **"2211Asn6F23.zip"**. Also included is the **"Edge.h"** file representing the "Edge" class. You should complete the implementation of the **"Graphs"** class.
- The zip file **"2211Asn6F23.zip"** contains a test graph file **"TestG.xlsx"** representing a graph of 7 vertices, as well as the corresponding Excel file. You can test your implementation using this file. The sample output for that file is also given in file **"Sample.txt"** in the zip file.
- The zip file also contains the graph file **"CitiesG.xlsx"** that will be used in the assignment.