Laboratory Class 4

Graphs: Adjacency Matrix Structure

Estruturas de Informação

A ***graph*** *G is* a set *V* of ***vertices*** and a collection *E* of pairs of vertices from *V*, called ***edges***. The aim of this worksheet is to complete and use an implementation of the Graph ADT based on the ***adjacency matrix*** ***r***epresentation.

As illustrated in figure 1, with this representation the set ***V* of *vertices*** are stored in an **ArrayList** and the set of edges are represented in a **matrix**.

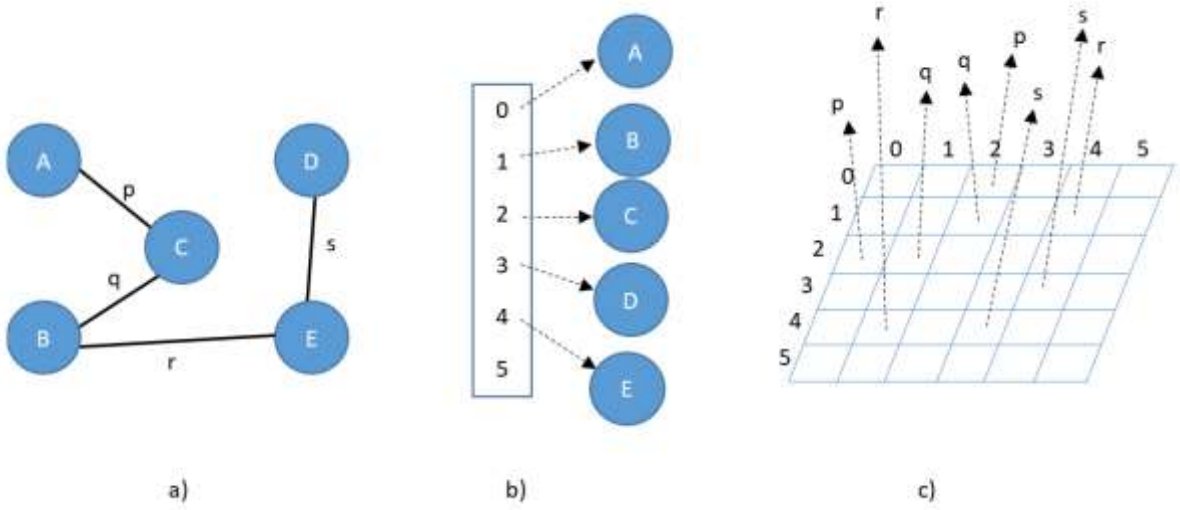

Figure1 - (a) An undirected graph G; (b) vertices' ArrayList; (c) edges' matrix

Download and extract the project **AdjacencyMatrixGraph_Initial** from moodle and analyse the classes.

All the classes use generic parameters V and E to designate the element type stored respectively at vertices and edges.

Part 1

1. Complete the generic class **AdjacencyMatrixGraph<V, E>** to include the methods to (i) return the set of outgoing edges of a particular vertex; and (ii) return the set of directed connected vertices from a particular vertex.
2. Complete the generic class **GraphAlgorithms<V,E>** to include methods to (i) return a BFS visit of the graph from a particular vertex; (ii) return all paths between a pair of vertices; and (iii) return a new graph which represents the transitive closure of the current graph.

3. Complete the generic class **EdgeAsDoubleGraphAlgorithms<V,E>** to include methods to (i) return the shortest path between a pair of vertices; and (ii) return a new graph which represents the minimum distances between each pair of vertices of the current graph.
4. Complete the test classes to test the missing implementation of the methods in 1. 2.iii and 3.ii.

Part 2

1. Implement the **HighwayMap** class which is intended to represent a map of cities and the interconnected highways. Each city has a name. Highways have name, distance (double) and cost (double).
   a. Declare the classes, and their attributes, using an adjacency map graph to represent the map (suggestion: create an external public class to represent a highway)
   b. Implement methods to insert cities and highways in the map.
   c. Implement methods to return the set of highways departing from a particular city and the set of cities connected through just one highway (one hop).
   d. Implement a method to check if two cities are reachable through highways.
   e. Implement a method to return the path between two cities using the minimum number of highways.
   f. Implement a method to return the path between two cities with the minimum distance.
   g. Implement a method to return the path between two cities with the minimum cost.

2. Create unit tests for methods developed in 1.e to 1.h.