**Practical Work 1**

**Graph Representation**

The implementation uses **two dictionaries** to store for each vertex the **list of inbound neighbours** and the **list of outbound neighbours**, respectively. Additionally another **dictionary** is used **for storing information** specific to every edge.

The **class DirectedGraph** will have the following data memebers:

*dict NeighboursIn*

dictionary of lists of inbound neighbours

*dict NeighboursOut*

dictionary of lists of outbound neighbours

*dict Costs*

dictionary of edges costs

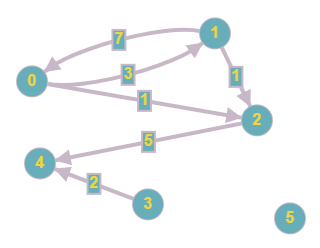
*int NumberOfEdges*

integer representing the number of edges of the graph

*int NumberOfVertices*

integer representing the number of vertices of the graph

**Examples to illustrate the representation**



Input:

6 6

0 1 3

0 2 1

1 0 7

1 2 1

2 4 5

3 4 2

The graph has 6 vertices and 6 edges. So graph.NumberOfVertices = **6** and graph.NumberOfEdges = **6**.

The dictionary of inbound neighbours will have the following structure:

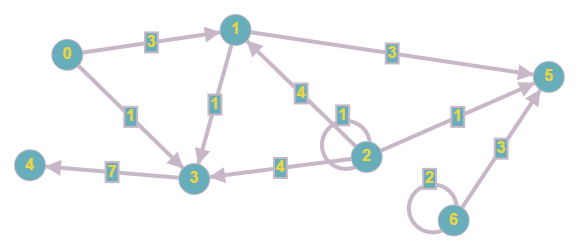
graph.NeighboursIn = **{ 0:[1], 1:[0], 2:[0, 1], 3:[ ], 4:[2, 3], 5:[ ] }**

The dictionary of outbound neighbours will have the following structure:

graph.NeighboursOut = **{ 0:[1, 2], 1:[0, 2], 2:[4], 3:[4], 4:[ ], 5:[ ] }**

The dictionary of costs will have the following structure:

graph.Costs = **{ (0, 1):3, (0, 2):1, (1, 0):7, (1, 2):1, (2, 4):5, (3, 4):2 }**



Input:

7 11

0 1 3

0 3 1

1 3 1

1 5 3

2 1 4

2 2 1

2 3 4

2 5 1

3 4 7

6 5 3

6 6 2

The graph has 7 vertices and 11 edges. So graph.NumberOfVertices = **7** and graph.NumberOfEdges = **11**.

The dictionary of inbound neighbours will have the following structure:

graph.NeighboursIn = **{ 0:[ ], 1:[0, 2], 2:[2], 3:[0, 1, 2], 4:[3], 5:[1, 2, 6], 6:[6] }**

The dictionary of outbound neighbours will have the following structure:

graph.NeighboursOut = **{ 0:[1, 3], 1:[3, 5], 2:[1, 2, 3, 5], 3:[4], 4:[ ], 5:[ ], 6:[5, 6] }**

The dictionary of costs will have the following structure:

graph.Costs = **{ (0, 1):3, (0, 3):1, (1, 3):1, (1, 5):3, (2, 1):4, (2, 2):1, (2, 3):4, (2, 5):1, (3, 4):7,**

**(6, 5):3, (6, 6):2 }**