# **Specification**

We will define two classes: **Graph** and ***WeightedGraph***, representing a Directed Graph and a Weighted Directed Graph, respectively. We will use two lists of neighbors for each vertex, to facilitate the parsing of both inbound and outbound edges for a given vertex. In addition, the WeightedGraph will contain a dictionary which will map each edge to a cost (real value).

Each vertex is uniquely identified by an integer denoting its index, and each edge is denoted by the two vertices which it unites.

The class ***Graph*** provides the following public methods and fields:

# **\_\_init\_\_()**

Creates a new, empty graph

# **loadGraph()**

Loads a graph, and its number of vertices/ edges, from a text file

# **saveGraph()**

Writes a graph (and its number of vertices/ edges) to a text file

# **getVerticesCount()**

Determines the number of vertices in the graph

# **getVertices()**

Iterates over the vertices in the graph

# **isEdge(srcVertex, destVertex)**

Checks if an edge from srcVertex to destVertex exists

# **getInDegree(vertex)**

Determines the in degree of vertex (how many edges end in that vertex)

# **getOutDegree(vertex)**

Determines the out degree of vertex (how many edges start in that vertex)

# **getInEdges(vertex)**

Iterates over the inbound edges of vertex

# **getOutEdges(vertex)**

Iterates over the outbound edges of vertex

# **addEdge(srcVertex, destVertex)**

Adds a new edge from srcVertex to destVertex to the graph

# **removeEdge(srcVertex, destVertex)**

Removes the edge from srcVertex to destVertex from the graph

# **addVertex()**

Adds a new vertex to the graph

# **removeVertex(index)**

Removes the vertex with the given index

# **createCopy()**

Creates a copy of the current graph

# **generateRandomGraph(nrVertices, nrEdges)**

Generates a random graph with a given number of vertices and edges

The class ***WeightedGraph*** is inherited from ***Graph*** and provides the following public methods and fields (in addition to those in ***Graph***):

# **\_\_init\_\_()**

Creates a new, empty weighted graph (overwrites the method from ***Graph***)

# **loadGraph()**

Loads a weighted graph from a text file (overwrites the method from ***Graph***)

# **addEdge(srcVertex, destVertex, cost)**

Adds a new edge from srcVertex to destVertex, with a given cost, to the graph (overwrites the method from ***Graph***)

# **removeEdge(srcVertex, destVertex)**

Removes the edge from srcVertex to destVertex from the graph (overwrites the method from ***Graph***)

# **removeVertex(index)**

Removes a given vertex from the graph (overwrites the method from ***Graph***)

# **getEdgeCost(srcVertex, destVertex)**

Determines the cost of the edge from srcVertex to destVertex

# **modifyEdgeCost(srcVertex, destVertex, newCost)**

Modifies the cost of the edge from srcVertex to destVertex

# **saveGraph()**

Writes the graph to a text file (overwrites the method from ***Graph***)

# **generateRandomGraph(nrVertices, nrEdges)**

Generates a random graph with a given number of vertices and edges (overwrites the method from ***Graph***)

# **printGraph()**

Prints a graph to the screen

# **Implementation**

The class ***Graph*** provides the following private methods and fields:

# **\_isActiveVertex(index)**

Checks if the vertex with a given index exists (and has not been deleted)

# **\_newVertex(index)**

Creates a new vertex

# **\_initEmptyGraph()**

Initializes an empty graph

# **\_dictIn**

Dictionary which holds which vertices have an edge from them to the current vertex

# **\_dictOut**

Dictionary which holds which vertices have an edge from the current vertex to them

# **\_nrVertices**

Integer, the number of vertices currently in the graph

# **\_nrEdges**

Integer, the number of edges currently in the graph

# **\_nextVertex**

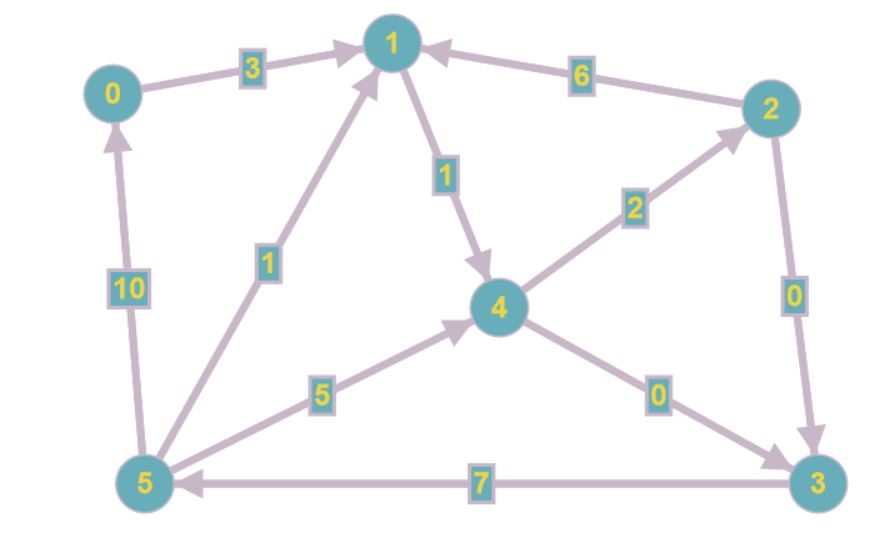
Integer, the index of the next vertex which will be added to the graph

The class ***WeightedGraph*** provides the following private methods and fields (in addition to those in ***Graph***):

# **\_initEmptyGraph()**

Initialises an empty weighted graph (overwrites the method from ***Graph***)

**\_dictCost** = dictionary which holds the cost of each edge of the graph



costEdges = {

(0, 1): 3

(1, 4): 1

(2, 1): 6

(2, 3): 0

(3, 5): 7

(4, 2): 2

outEdges = {

0: [1]

1: [4]

2: [1, 3]

3: [5]

4: [2, 3]

5: [0, 1, 4]

}

inEdges = {

(4, 3): 0

(5, 0): 10

(5, 1): 1

(5, 4): 5

}

0: [5]

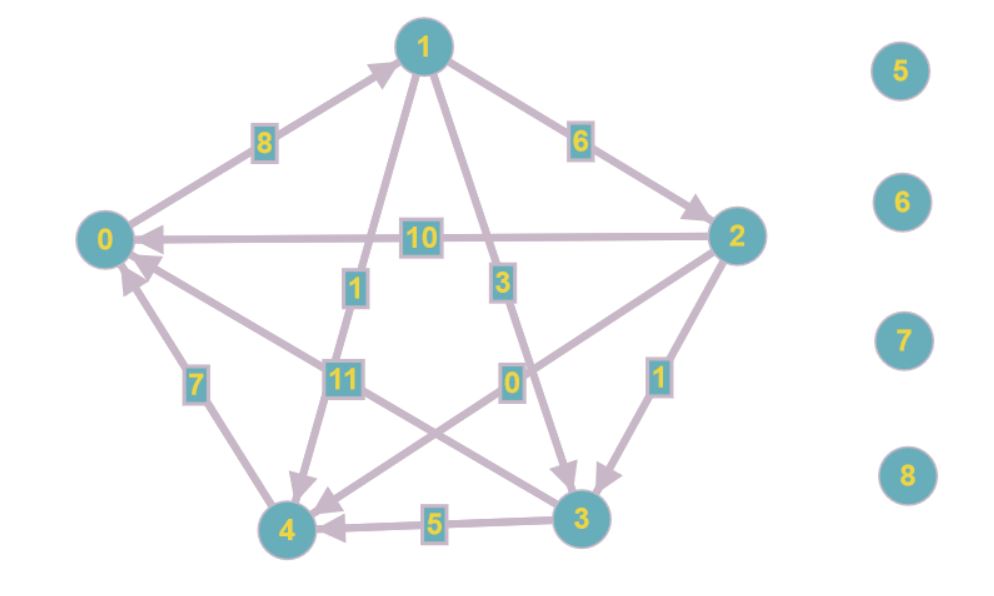
1: [0, 2, 5]

2: [4]

3: [2, 4]

4: [1, 5]

5: [3]

}

costEdges = {

(0, 1): 8

(1, 2): 6

(1, 3): 3

(1, 4): 1

(2, 10): 10

(2, 3): 1

(2, 4): 0

(3, 0): 11

(3, 4): 5

(4, 0): 7

}

outEdges = {

0: [1]

1: [2, 3, 4]

2: [0, 3, 4]

3: [0, 4]

4: [0]

5: []

6: []

7: []

8: []

}

inEdges = {

0: [2, 3, 4]

1: [0]

2: [1]

3: [1, 2]

4: [1, 2, 3]

5: []

6: []

7: []

8: []

}