**Practical Work no. 1**

**Fătu Samuel**

**Group 912**

**Specifications**

We shall define a class named Graph representing a directed graph.

The class Graph will provide the following methods:

Graph();

Constructs an empty graph

Graph(int noOfVertices);

Constructs a graph with noOfVertices vertices

Graph(const Graph& graph);

Copy constructor

bool is\_vertex(int vertex) const;

Returns true if the vertex is in the graph, false otherwise

bool is\_edge(int vertex0, int vertex1) const;

Returns true if (vertex0, vertex1) is an edge in the graph, false otherwise

int count\_vertices() const;

Returns the number of vertices

int count\_edges() const;

Returns the number of edges

int get\_in\_degree(int vertex) const;

Returns the in degree of a vertex

int get\_out\_degree(int vertex) const;

Returns the out degree of a vertex

void set\_edge\_data(int vertex0, int vertex1, int value);

Sets the cost of the edge (vertex0, vertex1) to value

int get\_edge\_data(int vertex0, int vertex1) const;

Gets the cost of the edge (vertex0, vertex1)

void add\_edge(int vertex0, int vertex1);

Adds an edge from vertex0 to vertex1, default cost is 0

void remove\_edge(int vertex0, int vertex1);

Removes the edge from vertex0 to vertex1

void add\_vertex(int vertex);

Adds the given vertex to the graph

void remove\_vertex(int vertex);

Removes the given vertex from the graph

std::unordered\_set<int>::iterator vertices\_begin() const;

Returns an iterator to the begining of the vertex set

std::unordered\_set<int>::iterator vertices\_end() const;

Returns an iterator to the end of the vertex set

std::unordered\_set<int>::iterator in\_begin(int vertex) const;

Returns an iterator to the begining of the inbound vertices of vertex

std::unordered\_set<int>::iterator in\_end(int vertex) const;

Returns an iterator to the end of the inbound vertices of vertex

std::unordered\_set<int>::iterator out\_begin(int vertex) const;

Returns an iterator to the begining of the outbound vertices of vertex

std::unordered\_set<int>::iterator out\_end(int vertex) const;

Returns an itera tor to the end of the outbound vertiecs of vertex

~Graph();

Destructor of the class

std::ostream& operator<<(std::ostream& os, const Graph& g);

Returns an ostream representing the graph

Graph readGraphFromFile(std::string filePath);

Reads a graph from the given file and returns it

void writeGraphToFile(Graph g, std::string filePath);

Writes the given graph the to given file

Graph createRandomGraph(int vertices, int edges);

Generates a random graph with `vertices` vertices and `edges` edges

**Implementation**

The implementation of the class uses an unordered set which represents the set of vertices and 3 unordered maps:

std::unordered\_map<int, std::unordered\_set<int>> inVertices;

Stores the in bound neighbours

std::unordered\_map<int, std::unordered\_set<int>> outVertices;

Stores the out bound neighbours

std::unordered\_map<std::string, int> verticesData;

Stores the cost of an edge