Assignment 1

Graph Algorithms

Language: Python

Structure:-a Graph class- contains all the methods and functionalities of the graph

-a UI class- used for user input/output and for calling fucntionalities from graph class

Graph class attribute s:

-self.\_in

-returns the dictionary of inbound vertices of the graph

-private(only accessible from inside the Graph or from outside using getters)

-self.\_out()

-returns the dictionary of outbound vertices of the graph

-private(only accessible from inside the Graph or from outside using getters)

-numberVertices

-the number of vertices of the graph

-numberEdges

-the number of edges of the graph

Graph class methods:

-dictionary\_cost(self)

-returns the dictionary of edges and costs of the graph

-private(only accessible from inside the Graph or from outside using getters)

-parse\_vertices(self)

-a list of all the vertices of the current graph

-parse\_inbound(self, x)

-a list of all the inbound vertices of the vertex x

-parse\_outbound(self, x)

- a list of all the outbound vertices of the vertex x

-parse\_cost(self)

-a list of all the costs of every edge of the graph

-add\_vertex(self, vertex)

-adds a new vertex to the graph

-remove\_vertex(self, vertex)

-removes a vertex from the graph

-in\_degree(self, vertex)

-returns the in degree of the vertex or FALSE if the vertex does not exist

-out\_degree(self, vertex)

-returns the out degree of the vertex or FALSE if the vertex does not exist

-add\_edge(self,vertex1, vertex2, cost)

-adds a new edge to the graph

-remove\_edge(self, vertex1, vertex2)

-removes an edge from the graph

-find\_if\_edge(self, vertex1, vertex2)

-returns the cost of the edge

-update\_cost(self, vertex1, vertex2, cost)

-changes the cost of an edge

-write\_graph\_to\_file(graph, file)

-writes in a file(of name chosen by the user) the graph

-read\_graph\_from\_file(file\_name)

-reads the graph from the file received as an input(fromthe user ex: graph1k/10k/100k/1m.txt)

Class UI attributes:

-graphs- a list of all the graphs

-current- the current graph(on which are applied the operations)

Class UI methods:

-create\_random\_graph\_ui(self)

-creates a random graph and adds it to the list of graphs in the program

-generate\_random(vertices, edges)

-generates a random graph

-read\_graph\_from\_file\_ui(self)

-takes the filename from the input and creates a new graph and adds it to the list of graphs

-write\_graph\_to\_file\_ui(self)

-writes the current graph that the program is working to a file chosen by user

-get\_number\_of\_vertices\_ui(self)

-prints the number of vertices of the current graph

-list\_outbound(self)

-prints the outbound vertices of the vertex received as an input

-list\_inbound(self)

-prints the inbound vertices of the vertex received as an input

-parse\_all\_vertices(self)

-prints all the vertices of the graph

-add\_vertex\_ui(self)

-adds the vertex received as an input to the graph

-delete\_vertex\_ui(self)

-deleted the vertex received as an input from the graph

-add\_edge\_ui(self)

-uses the add\_edge function to add an edge with a cost to the graph with the 2 vertices and the cost received as inputs

-remove\_edge\_ui(self)

-uses the remove\_edge function to delete an edge between the 2 vertices received as inputs

-update\_cost\_ui(self)

-uses the change\_cost function to change the cost of an edge between the vertices received as inputs

-in\_degree\_ui(self)

-prints the in degree of a vertex received as an input

-out\_degree\_ui(self)

-prints the out degree of a vertex received as an input

-check\_if\_edge\_ui(self)

-checks if the edge received as an input exists