**CME 2201 - Assignment 2**

**JOURNEY PLANNER FOR PARIS METRO**

**2021510122 Talha Mustafa Antep**

**2022510071 Mehmet Yörüdü**

Average Query Time of the code is 119.73 ms.

Vertex Class :

The class keeps a name or label for that point and remembers connections to other points. It keeps if its been visited, keeps the previous point on a path, and holds the weight of it.It has funcitons like label, edgeList (list of connections), visited, previousVertex, and cost. Methods like getLabel() give the point's name, and others like visit() or setCost() help manage if it's been visited and the cost to get there.

Edge’ s are in this class. This class is like a road that connects two points. It keeps track of which point is at the end of the road, how long the road is (weight), and what the road is called (routeName).

Edge Class (In the Vertex Class):

The Edge class represents a road between two points in a map. It holds information about which point is at the end of the road, weights of the road, and routeName. It has methods, getEndVertex() , getWeight() , getRouteName() , help find the end point, length of the road, and its name, respectively. This class helps manage connections between points in a map.

Directed Graph Class:

DirectedGraph class represents a structure for graphs. It stores nodes and their data in a DictionaryInterface structure. It keeps count of edges using the edgeCount variable.This class uses addVertex to add new nodes and addEdge to create directed connections between nodes. For checking if an edge exists between nodes, it offers the hasEdge method.To see if the graph is empty, isEmpty is used. To clear all nodes and edges, clear is used. It also includes getNumberOfVertices and getNumberOfEdges to count the nodes and edges in the graph.The resetVertices method resets node states, clearing visited status, costs, and previous nodes.Its main role is to perform basic graph operations like adding nodes, forming edges, and managing node and edge properties.The DirectedGraph class in the Test class represents a graph structure. It handles nodes and their data using a DictionaryInterface structure. This class also keeps track of the edge count using edgeCount.In the Test class, DirectedGraph uses addVertex to add new nodes and addEdge to create directed connections between nodes. It has the hasEdge method for checking if an edge exists between specified vertices.To check if the graph is empty, isEmpty is used. For clearing all nodes and edges, clear is used. It also includes getNumberOfVertices and getNumberOfEdges to count the nodes and edges in the graph.The resetVertices method in DirectedGraph resets node states, clearing visited status, cost, and predecessor node information.Overall, the DirectedGraph class in the Test class helps with basic graph tasks like adding nodes, creating edges, and managing properties of the graph.