ITCS-6114: Algorithm and Data Structure
Project 1: Shortest Path in a Network

In this project, I have worked on a program to find the shortest path in a network using the Dijkstra algorithm.

Language: Java
Data Structures: Hashmap, LinkedList, Queue, Graph, Heap
IDE: Eclipse

Running The Program
The program runs from the command-line:

- javac Graph.java is used to compile the program
- java Graph.java network.txt queries.txt output.txt

Here, network.txt is a file that contains the initial state of the graph. The graph created is a directed graph with two edges, one in each direction, for each input link. And queries.txt is a file that contains the queries that will be run on the graph. The queries come from the standard input and the output from the program goes to the standard output.The queries indicating changes to the graph are -

- addedge tailvertex headvertex transmit_time — Adds a single directed edge from tailvertex to headvertex.
- deleteedge tailvertex headvertex — Delete the specified directed edge from the graph. Does not remove the vertices. If the edge does not exist, we do nothing.
- edgedown tailvertex headvertex — Marks the directed edge as “down” and therefore unavailable for use.
- edgeup tailvertex headvertex — Marks the directed edge as “up”, and available for use.
- vertexdown vertex — Marks the vertex as “down”. None of its edges can be used.
- vertexup vertex — Mark the vertex as “up” again.
- path from_vertex to_vertex — This query is for finding the shortest path from_vertex to_vertex where from_vertex and to_vertex are names of vertices. This should compute the shortest time path from from_vertex to to_vertex using Dijkstra’s algorithm and based on the current state of the graph.
- Print — The simple query print must print the contents of the graph. Vertices must be printed in alphabetical order and the outward edge for each vertex must be printed in alphabetical order.
- Quit — The input query quit should simply cause the program to exit without printing anything.

All the output from the processed query is saved in a text file - Output.txt.

Program Summary
The program contains various classes that perform the required functions.Vertex class is for storing the vertexes of the Graph. Edge class stores the edges of the graph.

weightGraph contains all the main methods which are required to perform the queires on the Graph like addEdge, deleteEdge, edgeDown, vertexDown, edgeUp,vertexUp etc. minHeap class is for bulding the Binary heap and perform Dijkstra using heap operations. Function of Reachable class is based on the BFSearch algorithm.

Time Complexity:

Initialization - O(V)

BFS Running time - O(V+E)

For V vertex - O(V(V+E))