Name: Tushar Vijay Nemade
Student Id: 801257370

Project 1: Shortest Path in Network

Program Design:

Code is done in a single file named project1.Java. Aim of code is to perform a given task like generating and printing graphs, operations on graphs, findinding shortest path using Dijkstra's algorithm, printing reachable vertices. Dijkstra's algorithm is implemented using priority queue. Java inbuilt priority queue is used.

Classes defined:

vertexParam:

It is used to represent the vertex of the class. It has attributes like name, adjacent vertex list, vertex status, weight, previous vertex and visited. Constructor is used to set the default value of the vertex.By default weight is set as infinity.

edgeParam:

It is used to represent the edge of a graph.It has attributes like destination vertex of the vertexParam, edgeStatus, weight. Constructor is used to set the default value of edge like edgeStatus as UP for every edge initially.

Project1:

This class implements every functionality needed by the project. It consists of different functions which are used to implement this functionality.

Tasks:

Functions implemented:

Task 1: build graph

createVertex(String vertexName):

This function is used to create the vertex and add it to hashmap. Here a hashmap is used to store the vertex and its data. Input to this function is vertexName.

createEdge(String source, String destination, double weight):

Functionality**:** Add edge to graph

This function is used to add edges to the graph. In this function destination vertex is added in the source vertex adjacent list. Along with the edge its weight is also updated. First source vertex is fetched from vertexMap.

Task 2: print graph

printGraph():

This function is used to print the basic graph which is built by createVertex and createEdge functions. In this every vertex and its adjacent vertices are printed alphabetically along with its weight and vertex status and edge status if it is down.

Task 3: find shortest path

implementDjkstras(String source) and printPath():

This class is used to find the shortest path from source to destination. It uses Dijkstra's algorithm using priority queue to implement the shortest path. It takes the

source vertex, visits adjacent vertices and adds each adjacent vertex to queue and compare the distances and update the distance whichever is minimum. Then the queue is iterated until it becomes empty and at last we get the shortest path. The priority queue is minimum weight priority queue means it removes the adjacent vertex which has minimum weight. printPath is used to print the shortest path generated by Dijstras algorithm.

Task 4: Operations on graph/update graph
changeVertexStatus(String source):
Functionality: vertex Up and vertex Down

This function is used to change the status of vertex UP or Down. It just updates the vertexStatus attribute in the vertexParam class for the vertex.

changeEdgeStatus(String source, String dest) and checkEdgeStatus(source,dest):
Functionality: edge Up and edge Down

This function is used to change the status of edges. It fetches the source vertex from vertexMap and changes the status of the destination vertex from source vertex adjacency list. It changes the edgeStatus flag from Up to down or vice versa. checkEdgeStatus function returns boolean value true if edge is down and vice versa.

deleteEdge(source,dest):
Functionality: Delete edge

This function removes the edge from the graph, that is the edge no longer exists in the graph for traversal. It fetches the source vertex from vertexMap and then finds the destination vertex in its adjacency list and removes it from the list.

Task 5: Reachable vertices
ReachableVertices(vertexParam V):

This function is used to find the reachable vertices from the given vertices and print them.

Compiler Details:

Compiler: javac.
Version: 11.0.11

Command to compile: javac Project1.java after execution creates all .class files.
Command to run: java Project1 inputfile.txt

Data Structure Used:

- **HashMap:** It is used to store the vertices and its details.
- **TreeMap:** It is used to print the vertices alphabetically in graphprint function. As TreeMap store keys in sorted order. Has overridden its sort function using comparator as ordering is done on variables inside the object.

Name: Tushar Vijay Nemade
Student Id: 801257370

- **Treeset:** It is also used to print vertices alphabetically in the printReachable function. The treeset also stores values in sorted order.
- **Linked list:** It is used to store the adjacent vertices of the vertex and details of adjacent vertices. That is the details of the edges of the graph
- **Priority Queue:** It is used to implement the dijkstra's algorithm. It is used to find the shortest weight vertex in the queue.

Reachable Algorithm:

Complexity: O(V + E)

```
reachableVertices(v)
    Foreach vertex in v.adjacancyList
        If vertex.dest.status not down
            Continue
        If edge not down
            Continue
        If vertex.dest.color==”white”
            Add vertex.dest to reachableMap
            vertex.dest.color=”black”
            reachableVertices(vertex.dest)
```

Complexity: O(V *(V + E))

```
printReachable()
    For Each vertex in vertexMap
        Add vertex to treemap
    Foreach vertex in treemap
        If vertex NOT down
            then reachableVertices(vertex)
            Print treeset
            Clear treeset
```

Time complexity:

As each vertex is visited and all its adjacent vertices are also visited which denotes the edges. And as printReachable is called for every vertex. So complexity is **O(V *(V + E)).**
The algorithm works nearly the same as the DFS algorithm.

What works and Fails?

- While reading the input file if the source destination and weight is not given properly it will affect the building of the graph. For e.g : if only source and destination is given but weight is not given.
- All the five tasks that are build graph, print graph, find shortest path, update graph and reachable vertices are working properly.
- For users to run the code they need to know the exact queries. If they don’t then they will not be able to run the code.