REPORT

IT-427

PROG ASSIGNMENT – 2

Question- Write a java code to find the shortest path of a given graph(file) using Dijkstra’s algorithm.

The commands to run the code for the given file

wdGraphs.txt

Compile – javac dijkstra.java

Run – java dijkstra wdGraphs.txt

The following is the output-

**Summary of my code**

In my code, I have created the public class named Dijkstra which contains the main method where I am reading the input file name from the command line arguments

I have created a Dijkstra class and a Node class

In Dijkstra class I am having all the code for parsing and calculating shortest path using Dijkstra algorithm.

The Node class contains a vertex and distance which helps in storing the graph Data

**Methods used-**

parseFile (return type - void)

* This method takes filename as the input and parses the file line by line  
  and in this process I am collecting the graph data into a Map and once the graph is ready then I am calling another method called **calculateDijkstra.**

calculateDijkstra(return type - void)

* This method takes the graph, graphSize i.e no of vertices, a 2D array which is a representation of the edges, and a list of valid nodes(which indicates the nodes having atmost 1 edge)
* I used Dijkstra’s algorithm to solve the graph i.e by by greedy method and found the shortest path from source to each node
* I am keeping track of the best path to end vertex and traversing it in order to get the shortest path track and printing the output

**Data Structures Used-**

* List
* Map

**Algorithm**- I used Dijkstra’s algorithm to solve this problem i.e via greedy method

I am defining the initial distance as DOUBLE.MAX\_VALUE (infinity) and starting at 0 vertex as source and calculating the cost to reach the neighboring vertices and updating the weights of that nodes accordingly and after that I am adding those nodes into a priority queue (which maintains the nodes in ascending order) by this method when we reach the final vertex we will have the shortest calculated path.

Time complexity of my code-

Using priority queue the complexity of the algorithm will be **O(|E|+|V|log|V|)** where ,

**E** is the number of edges

**V**  is the number of nodes.

Space complexity of my code-

My code has 2 for loops- and the worst case will be O(V\*V)

Where V is the number of vertices of the graph