PRACTICAL – 5

AIMAN SIDDIQUA 2K18/MC/008

AIM: To write a program to find the shortest path between two vertices in a graph using Dijkstra’s Algorithm.

CODE:

#include <bits/stdc++.h>

**using** **namespace** std;

#define V 8

**int** **minDistance**(**int** dist[], **bool** sptSet[])

{

**int** min = INT\_MAX, min\_index;

**for** (**int** v = **0**; v < V; v++)

**if** (sptSet[v] == false && dist[v] <= min)

min = dist[v], min\_index = v;

**return** min\_index;

}

**void** **dijkstra**(**int** graph[V][V], **int** src)

{

**int** dist[V];

**bool** sptSet[V];

**for** (**int** i = **0**; i < V; i++)

dist[i] = INT\_MAX, sptSet[i] = false;

dist[src] = **0**;

**for** (**int** count = **0**; count < V - **1**; count++) {

**int** u = minDistance(dist, sptSet);

sptSet[u] = true;

**for** (**int** v = **0**; v < V; v++)

**if** (!sptSet[v] && graph[u][v] && dist[u] != INT\_MAX

&& dist[u] + graph[u][v] < dist[v])

dist[v] = dist[u] + graph[u][v];

}

**for** (**int** i = **0**; i < V; i++)

cout <<"Distance of vertex " << i << " from source is "<<dist[i]<< endl;

}

**int** **main**()

{

**int** graph[V][V] = { { **0**, **8**, **0**, **0**, **0**, **0**, **0**, **4**},

{ **8**, **0**, **4**, **0**, **0**, **0**, **0**, **1**},

{ **0**, **7**, **0**, **4**, **0**, **8**, **0**, **0**},

{ **0**, **0**, **7**, **0**, **9**, **4**, **0**, **0**},

{ **0**, **0**, **0**, **9**, **0**, **10**, **0**, **0**},

{ **0**, **0**, **4**, **11**, **10**, **0**, **2**, **0**},

{ **0**, **0**, **0**, **0**, **0**, **2**, **0**, **1**},

{ **8**, **6**, **0**, **0**, **0**, **0**, **11**, **0**}};

dijkstra(graph, **0**);

**return** **0**;

}

OUTPUT:

