**ASSIGNMENT 4**

**AIM:**

For a weighted graph G, find the minimum spanning tree using Prims Algorithm.

**CODE:**

#include<iostream>

using namespace std;

void create(int mat[][10], int v)

{

int v1,v2,cost;

int edges;

cout<< "\nEnter the total number of edges : ";

cin>> edges;

for(inti=0;i<v;i++)

{

for(int j=0;j<v;j++)

{

mat[i][j] = 0;

}

}

for(inti=0;i<edges;i++)

{

cout<< "\nEnter edge : ";

cin>> v1 >> v2;

cout<< "\nEnter the cost of that edge : ";

cin>> cost;

mat[v1][v2] = cost;

}

}

voiddisplay\_matrix(int mat[][10],int v)

{

cout<< "\nAdjacency matrix representation :- " <<endl;

for(inti=0;i<v;i++)

{

for(int j=0;j<v;j++)

{

cout<< mat[i][j] << "\t";

}

cout<<endl;

}

}

intmin\_dist(intdist[],int visited[],int v)

{

int min = 32767;

intmin\_index;

for(inti=0;i<v;i++)

{

if(visited[i] == 0 &&dist[i] <= min)

{

min = dist[i];

min\_index = i;

}

}

cout<<min\_index<<endl;

returnmin\_index;

}

voidprint\_dist(int mat[][10],intdist[],intv,int parent[])

{

int sum=0;

cout<<"\nPRIM'S MST OF THE GRAPH IS: ";

for(inti = 1; i<v; i++)

{

cout<<"\n"<<i<<"-"<<parent[i];

sum = sum + mat[parent[i]][i];

}

cout<<endl;

cout<<"\nCOST OF MST IS: "<<sum<<endl;

}

void prims(int mat[][10],ints,int v)

{

intdist[v];

int visited[v];

int parent[v];

for(inti=0;i<v;i++)

{

dist[i] = 32767;

visited[i] = 0;

}

dist[s] = 0;

int p=0;

for(int j=0;j<v-1;j++)

{

p = min\_dist(dist,visited,v);

visited[p] = 1;

for(int q=0;q<v;q++)

{

if(mat[p][q] != 0)

{

if(visited[q] == 0 && mat[p][q] <dist[q])

{

dist[q] = mat[p][q];

parent[q] = p;

}

}

}

}

print\_dist(mat,dist,v,parent);

}

int main()

{

int v;

int s;

cout<< "\nEnter the number of vertices : ";

cin>> v;

int mat[v][10];

create(mat,v);

display\_matrix(mat,v);

cout<< "\nEnter source vertex : ";

cin>> s;

prims(mat,s,v);

return 0;

}

