Lab Experiment No.7 (prims\_algo)

-----------------------------------------------------------------------------------------------------------------------------

Name: Prathamesh Sadashiv Gadekar

Roll No.:14

Class : SE-IT

-----------------------------------------------------------------------------------------------------------------------------

Represent a graph of your college campus using adjacency list /adjacency matrix. Nodes should represent the various departments/institutes and links should represent the distance between them. Find minimum spanning tree

b) Using Prim’s algorithm.

#include<iostream>

#include<string>

#define SIZE 10

#define INFINITY 999

using namespace std;

class MST

{

int graph[SIZE][SIZE],vertex,v1,v2;

public:

MST()

{

for(int i=0;i<SIZE;i++) //initialise graph

{ for(int j=0;j<SIZE;j++)

graph[i][j]=0; }

}

void find\_PrimsMST()

{

int select[SIZE],i,j,k,min\_dist,total=0;

for(i=0;i<vertex;i++)

select[i]=0;

cout<<"\n The Minimum Spanning Tree ";

select[0]=1;

for(k=1;k<vertex;k++)

{

min\_dist=INFINITY;

for(i=0;i<vertex;i++)

{

for(j=0;j<vertex;j++)

{

if(graph[i][j] && ((select[i] && !select[j]) || (!select[i] && select[j])))

{

if(graph[i][j] <min\_dist)

{

min\_dist=graph[i][j];

v1=i;

v2=j;

}

}

}

}

cout<<" \n Edge (" <<v1 <<" " <<v2<<")"<<min\_dist;

select[v1]=select[v2]=1;

total=total+min\_dist;

}

cout<<"\n Total Distance: "<<total;

cout<<"\n";

}

void get\_data()

{

int len,n;

cout<<"\n Enter number of departments in the college ";

cin>>vertex;

cout<<"\n Enter number of edges :";

cin>>n;

cout<<"\n Enter edges and weights ";

for(int i=0;i<n;i++)

{

cout<<"\n Enter Edge by V1 and V2: ";

cin>>v1>>v2;

cout<<"\n Enter Corresponding Distance between two vertex : ";

cin>>len;

graph[v1][v2]=graph[v2][v1]=len;

}

}

void printmatrix()

{

cout<<"\n Adjacency Matrix for departments in college :\n";

for(int i=0;i<vertex;i++)

{for(int j=0;j<vertex;j++)

cout<<" "<<graph[i][j];

cout<<"\n";

}

} };

int main()

{

MST mt;

cout<<"\n\n Prim's Algorithm ";

mt.get\_data();

mt.printmatrix();

cout<<"\n \n";

mt.find\_PrimsMST() }

**OUTPUT :**

Prim's Algorithm

Enter number of departments in the college 4

Enter number of edges :5

Enter edges and weights

Enter Edge by V1 and V2: 1 2

Enter Corresponding Distance between two vertex : 11

Enter Edge by V1 and V2: 2 3

Enter Corresponding Distance between two vertex : 18

Enter Edge by V1 and V2: 3 4

Enter Corresponding Distance between two vertex : 12

Enter Edge by V1 and V2: 4 1

Enter Corresponding Distance between two vertex : 14

Enter Edge by V1 and V2: 1 3

Enter Corresponding Distance between two vertex : 10

Adjacency Matrix for departments in college :

0 0 0 0

0 0 11 10

0 11 0 18

0 10 18 0

The Minimum Spanning Tree

Edge (1 3)999

Edge (1 2)11

Edge (1 2)999

Total Distance: 2009