

Roll Number: SYCOC303

Division: C

PRN Number: 122B2B303

Batch: C4

Name: VINAYAK MADAN SHETE

Problem Statement:

⇒ Write a C++ program to find the minimum spanning tree of a given undirected graph.

INPUT:

```
/*
 * =====
 *      Program Name: MinSPT.cpp
 *      Created on: December 20, 2022
 *      Author: Vinayak Shete
 * =====
 */

#include<iostream>
#define infinity 9999

using namespace std;

class MinSPT
{
    int vertex;
    int Edges;
    int adj[100][100];
    int sp[100][100];
    int minCost;
    public:
        //constructor for assigning initial values in a matrix
        MinSPT(int V)
```

```
{
vertex = V;
for(int i=0;i<V;i++)
{
    for(int j=0;j<V;j++)
    {
        adj[i][j]=0;
        sp[i][j]=0;
    }
}

//function for getting minimum cost edge
int getMinCost()
{
    return this->minCost;
}

//function for creating a graph
void createGraph()
{
    int source, destination, weight;
    int ch;
    cout<<"The vertices from 0 to "<<vertex-1<<" are created for the graph!";
    cout<<"\nEnter the values for source and destination and the weight of
that edge-->";
    do
    {
        cout<<"\nEnter the source,destination and weight for an
edge:";
        cin>>source>>destination>>weight;
        adj[source][destination] = weight;
        adj[destination][source] = weight;
        cout<<"\nDo you want to add another edge?[1 for YES || 0 for No]-->";
        cin>>ch;
    }
    while(ch==1);
}
```

```
}

void mst()
{
    int cost[vertex][vertex]={0};
    int visited[10]={0};
    int distance[vertex] = {infinity};
    int source[vertex] = {0};
    int minDist=0;

    for(int i=0;i<vertex;i++)
    {
        for(int j=0;j<vertex;j++)
        {
            if(adj[i][j]==0)
                cost[i][j] = infinity;
            else
                cost[i][j] = adj[i][j];
        }
    }

    distance[0]=0;
    visited[0] = 1;
    int source_vertex, dest_vertex=0;

    for(int i=0;i<vertex;i++)
    {
        distance[i] = cost[0][i];
        source[i] = 0;
    }

    minCost=0;
    Edges=vertex-1;
    while(Edges>0)
    {
        minDist = infinity;
        for(int i=0;i<vertex;i++)
```

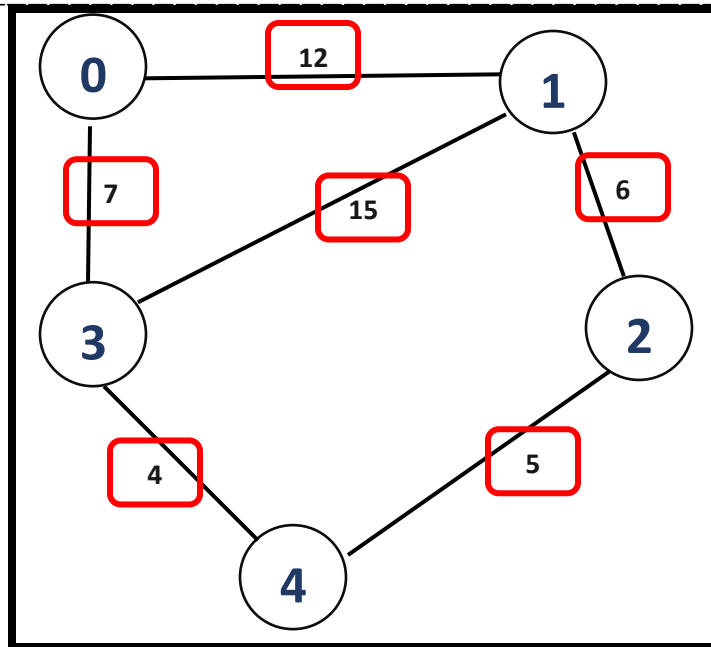
```
        {
            if(visited[i]==0 && distance[i]< minDist)
            {
                minDist= distance[i];
                dest_vertex= i;
            }
        }
        source_vertex=source[dest_vertex];
        sp[source_vertex][dest_vertex] =
sp[dest_vertex][source_vertex] = distance[dest_vertex];
        visited[dest_vertex] = 1;
        minCost+=cost[source_vertex][dest_vertex];
        Edges--;
        for(int i=0;i<vertex; i++)
        {
            if(visited[i]==0 && cost[dest_vertex][i]<
distance[i])
            {
                distance[i]= cost[dest_vertex][i];
                source[i]=dest_vertex;
            }
        }
    }
}

};

int main()
{
    int vert;
    cout<<"\n=====WELCOME===== ";
    cout<<"\nHow many vertices you want in the GRAPH? ";
    cin>>vert;
    MinSPT spObj(vert);
    //calling function to create a graph with the given number of vertices
    spObj.createGraph();
    spObj.mst();
    cout<<"\n===== ";
```

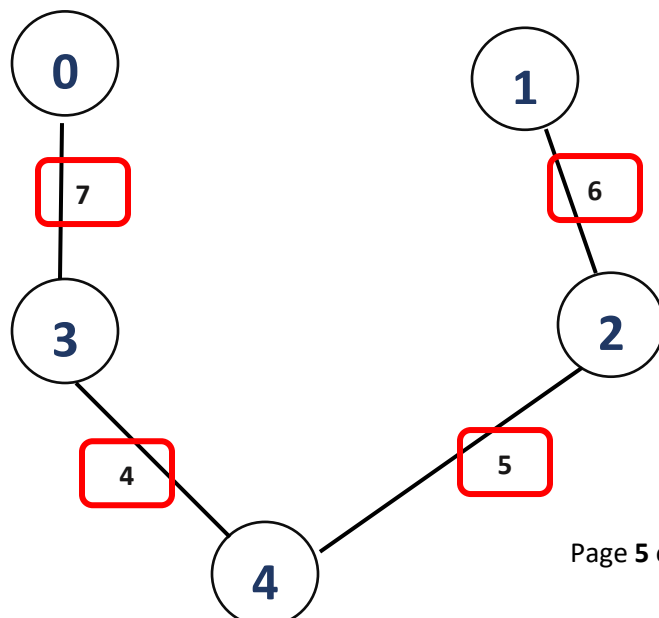
```
cout<<"\n\nThe cost of minimum spanning tree is: "<<spobj.getMinCost();  
cout<<"\n=====";  
while(true)  
{  
    }  
    cout<<"\n=====THANK YOU!=====";  
    return 0;  
}
```

OUTPUT:



Designing the Graph as shown in the above figure.

For the above graph minimum spanning tree will be:



The weight for this minimum spanning tree will be 22

```
=====WELCOME=====
How many vertices you want in the GRAPH? 5
The vertices from 0 to 4 are created for the graph!
Enter the values for source and destination and the weight of that edge-->
Enter the source,destination and weight for an edge:0 1 12

Do you want to add another edge?[1 for YES || 0 for No]-->1

Enter the source,destination and weight for an edge:0 3 7

Do you want to add another edge?[1 for YES || 0 for No]-->1

Enter the source,destination and weight for an edge:1 2 6

Do you want to add another edge?[1 for YES || 0 for No]-->1

Enter the source,destination and weight for an edge:1 3 15

Do you want to add another edge?[1 for YES || 0 for No]-->1

Enter the source,destination and weight for an edge:2 4 5

Do you want to add another edge?[1 for YES || 0 for No]-->1

Enter the source,destination and weight for an edge:3 4 4

Do you want to add another edge?[1 for YES || 0 for No]-->0

=====

The cost of minimum spanning tree is: 22
=====
```

=====