Name – Amit Bandu Swami

Roll No – 2221018

Ass 5

Problem :- You have a business with several offices; you want to lease phone lines to connect them up with each other; and the phone company charges different amounts of money to connect different pairs of cities. You want a set of lines that connects all your offices with a minimum total cost. Solve the problem by suggesting appropriate data structures.

//Code

```
#include<iostream>

using namespace std;

class Graph

{

        int ne,m,n,w;

        int cost[10][10];

        public:

                int nv;

                void create();

                void prims(int v,int visit[10]);

};

void Graph::create()

{

        int r,c; //r=row c=column
```

```
cout<<"Enter a no of vertices\t:"<<endl;
cin>>nv;
cout<<"\nEnter a no of edges \t:"<<endl;
cin>>ne;
for(int i=0;i<nv;i++)
{
        for(int j=0;j<nv;j++)
        {
                cost[i][j]=0;
        }
}
for(int i=0;i<ne;i++)
{
        cout<<"\nEnter the start and end vertex andweight:"<<endl;
        cin>>m>>n>>w;
        cost[m][n]=w;
}
cout<<"---------Adjacency matrix ------ "<<endl;
for(int i=0;i<nv;i++)
{
        for(int j=0;j<nv;j++)
        {
```

```
                    cout<<cost[i][j]<<" ";
                }
                cout<<endl;
        }
}
void Graph::prims(int v, int visit[10])
{
        int sum=0,p,temp,min,m;
        visit[v]=1;
        for(int k=0;k<nv-1;k++)
        {
                temp=999;
                for(int i=0;i<nv;i++)
        {
        if(visit[i]==1)
        {
        min=999;
        for(int j=0;j<nv;j++)
        {
                if(cost[i][j]!=0)
                {
                        if(min>=cost[i][j] && visit[j]==0)
                            {
                                      min=cost[i][j];
                                      p=j;
                            }
                }
        }
        if(min<temp)
        {
                temp=min;
                m=i;
                n=p;
        }
        }
        }
        sum=sum+cost[m][n];
        cout<<"selected edge between"<<m<<"to"<<n<<endl;
        visit[m]=1;
        visit[n]=1;
        }
        cout<<"minimum weight"<<sum<<endl;
```
```
}
int main()
```

```
{
Graph g1;
int ch;
int visit[10];
while(1)
{
        cout<<"1 for  create"<<endl;
        cout<<"2 for prims algorithm"<<endl;
        cout<<" 3 for exit"<<endl;
        cout<<"enter a your choice"<<endl;
        cin>>ch;
        switch(ch)
        {
                case 1:
                        g1.create();
                        break;
                case 2:
                        int v;
                        for(int i=0;i<g1.nv;i++)
                        {
                        visit[i]=0;
                        }
```

```
                    cout<<"enter a starting vertex"<<endl;
                    cin>>v;
                    g1.prims(v,visit);
                    break;
                case 3:
                    exit(0);
                    break;
                default:
                    cout<<"invalid choice"<<endl;
        }
        }
}
/*
output:
____MENU_____
1-Create
2-prims
3-Exit
--------------
enter your choice: 1
Enter a no of vertices :4
Enter a no of edges :5
Enter the start and end vertex and weight:0 1 1

Enter the start and end vertex and weight:1 2 4

Enter the start and end vertex and weight:2 3 6

Enter the start and end vertex and weight:3 1 5

Enter the start and end vertex and weight:0 3 2

---------Adjacency matrix-------

0 1 0 2

0 0 4 0

0 0 0 6

0 5 0 0

_____MENU_______

*/
```