BRACT's
VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY, PUNE – 48
An Autonomous Institute Affiliated to Savitribai Phule Pune University, Pune

SD(LP-II) ASSIGNMENT (S.Y.B. Tech. – DIV: C)

Name: Prajwal Dabhade;

Roll no.: 223012;

GR. No.: 17u385; Batch: C1

**********Assignment No.5**********

- **Aim**: 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 pair 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.
- **Objective:** We have to implement this using Minimum Spanning Tree with the use of graph data structure.
- **Theory**: Given a connected and undirected graph, a *spanning tree* of that graph is a subgraph that is a tree and connects all the vertices together. A single graph can have many different spanning trees. A *minimum spanning tree (MST)* or minimum weight spanning tree for a weighted, connected and undirected graph is a spanning tree with weight less than or equal to the weight of every other spanning tree. The weight of a spanning tree is the sum of weights given to each edge of the spanning tree.

Applications:

a) Building a connected network.
b) Clustering.
c) Traveling salesman problem.
d) Image registration and segmentation.

- **Program:**

```
#include<iostream>

using namespace std;

struct element

{

   int start[20],endp[20],cost[20];
```

```
};
class prims
{
public:
   int vertex,edge,matrix[10][10];
   string r[10];
   element b;
   prims()
   {
      r[0]="Pune";
      r[1]="Mumbai";
      r[2]="Nagpur";
      r[3]="Nashik";
      r[4]="Thane";
      r[5]="Alibag";
   }
   void input()
   {
      int i;
      for(i=0;i<edge;i++)
      {
         cout<<"\nEnter data:";
         cin>>b.start[i]>>b.endp[i]>>b.cost[i];
      }
```

```
}
void cost_matrix()
{
   int i,j;
   for(i=0;i<edge;i++)
   {
      for(j=0;j<edge;j++)
      {
         matrix[i][j]=999;
      }
   }
   for(i=0;i<edge;i++)
   {
      matrix[b.start[i]][b.endp[i]]=b.cost[i];
   }

}
 void display()
{
   int i,j;
   for(i=0;i<edge;i++)
   {
      cout<<b.start[i]<<"\t"<<b.endp[i]<<"\t"<<b.cost[i];
      cout<<"\n";
```

```
    }
    cout<<"\nMatrix is";
     for(i=0;i<edge;i++)
       {
          cout<<"\n";
          for(j=0;j<edge;j++)
             {
                cout<<matrix[i][j]<<"\t";
             }
       }

}

  void compute()
{
    int visited[vertex];
    visited[0]=1;
    int c=0,i,j,s=0;
    for(i=1;i<vertex;i++)
       visited[i]=0;
while (c<vertex-1)
{
   int min=9999,x=0,y=0;
  for (int i = 0; i<vertex; i++)
    {
```

```
    if (visited[i]==1)
    {
       for (int j = 0; j <vertex; j++)
       {
          if (visited[j]==0 && matrix[i][j])
          {
            if (min > matrix[i][j])
              {
                min = matrix[i][j];
                x = i;
                y = j;
              }
          }
       }
    }
    s=s+matrix[x][y];
     cout <<r[x] <<  " - " << r[y] << " :  " << matrix[x][y];
     cout << endl;
     visited[y]=1;
     c++;
    }
    cout<<"The total money required:"<<s<<endl;
  }
  }
};
```

```
int main()
{
   prims a;
   cout<<"\nEnter number of vertex and edges:";
   cin>>a.vertex>>a.edge;
   a.input();
   a.cost_matrix();
   a.display();
   a.compute();
}
```

- **Output:**

```
"E:\S.Y.B.Tech\SEM-II\Skill Development\Assignment5.exe"

Enter number of vertex and edges:3 3

Enter data:0 1
9

Enter data:1 2
5

Enter data:0 2
4
0       1       9
1       2       5
0       2       4

Matrix is
999     9       4
999     999     5
999     999     999     Pune - Nagpur :  4
Pune - Nagpur :  4
Pune - Nagpur :  4
The total money required:12

Process returned 0 (0x0)   execution time : 25.128 s
Press any key to continue.
```

- **Conclusion:** Thus, we have implemented minimum spanning tree using graph data structure.