

Data Structures and Algorithms

CSE2001

Lab - 7 - Assignment - 3

Yashwanth Reddy

19BCE7362

Date- 20thJuly2021

Problem : Write a Program to Implement Prim's Minimum Spanning Tree

Code

```
import java.util.*;

public class Prim {
    public int isVisited[] = new int[15];
    public int cost[][] = new int[10][10];
    public int minimum_cost;

    public void calc(int n)
    {
        int flag[] = new int[n+1];
        int i,j,min=999,num_edges=1,a=1,b=1,minpos_i=1,minpos_j=1;

        while(num_edges < n)
        {

            for(i=1,min=999;i<=n;i++)
                for(j=1;j<=n;j++)
```

```

        if(this.cost[i][j]<min)
            if(this.isVisited[i]!=0)
            {
                min=this.cost[i][j];
                a=minpos_i=i;
                b=minpos_j=j;
            }
            if(this.isVisited[minpos_i]==0 || this.isVisited[minpos_j]==0)
            {
                System.out.println("Edge Number \t"+num_edges+"\t from Vertex
\t"+a+"\t to Vertex \t"+b+"-mincost:"+min+" \n");
                this.minimum_cost=this.minimum_cost+min;
                num_edges=num_edges+1;
                this.isVisited[b]=1;
            }
            this.cost[a][b]=this.cost[b][a]=999;

    }

}

public static void main(String args[])
{
    int nodes,i,j;
    Scanner in = new Scanner(System.in);
    System.out.println("Enter the Number of Nodes \n");
    nodes = in.nextInt();
    Prim p = new Prim();
    System.out.println("Enter the Cost Matrix Weights : \n");
    for(i=1;i<=nodes;i++)

```

```

        for(j=1;j<=nodes;j++)
        {
            p.cost[i][j]=in.nextInt();
            if(p.cost[i][j]==0)
                p.cost[i][j]=999;
        }

        p.isVisited[1]=1;
        p.calc(nodes);
    }
}

```

Output

```

C:\Users\yashw\Desktop\Summer\Labs>java Prim
Enter the Number of Nodes
5
Enter the Cost Matrix Weights :
0 2 0 6 0
2 0 3 8 5
0 3 0 0 7
6 8 0 0 9
0 5 7 9 0
Edge Number    1      from Vertex    1      to Vertex    2-mincost:2
Edge Number    2      from Vertex    2      to Vertex    3-mincost:3
Edge Number    3      from Vertex    2      to Vertex    5-mincost:5
Edge Number    4      from Vertex    1      to Vertex    4-mincost:6

```