DHRUV GARG 16 BCE 1190

Data Structures and Algorithms

LAB 6 – MST by Kruskal's strategy

KRUSKAL'S STRATEGY

Code for MST using Kruskal's strategy

```
#include<stdio.h>
#include<stdlib.h>
int i,j,k,a,b,u,v,n,ne=1;
int min,mincost=0,cost[9][9],parent[9];
int find(int);
int uni(int,int);
void main()
{
   printf("\n\tIMPLEMENTATION USING
KRUSKAL'S ALGORITHMS\n");
   printf("\nNumber of vertices : ");
   scanf("%d",&n);
   printf("\nEnter adjacency matrix : \n");
   for(i=1; i<=n; i++)
   {
      for(j=1; j<=n; j++)
      {
         scanf("%d",&cost[i][j]);
         if(cost[i][j]==0)
            cost[i][j]=999;
      }
   }
```

```
    printf("\nEdges of the Minimum Spanning Tree ( MST ) are : \n");
    while(ne < n)
    {
        for(i=1,min=999; i<=n; i++)
        {
            for(j=1; j <= n; j++)
            {
                if(cost[i][j] < min)
                {
                    min=cost[i][j];
                    a=u=i;
                    b=v=j;
                }
            }
        }
        u=find(u);
        v=find(v);
        if(uni(u,v))
        {
            printf("%d Edge (%d,%d) = %d\n",ne++,a,b,min);
            mincost +=min;
        }
        cost[a][b]=cost[b][a]=999;
    }
    printf("\nMinimum cost = %d\n",mincost);
}
int find(int i)
{
    while(parent[i])
        i=parent[i];
    return i;
}
```

```
int uni(int i,int j)
{
  if(i!=j)
  {
    parent[j]=i;
    return 1;
  }
  return 0;
}
```

Screenshot for MST using Kruskal's method

C:\Users\Dhruv\Downloads\kruskal.exe

```
        IMPLEMENTATION USING KRUSKAL'S ALGORITHMS

Number of vertices : 6

Enter adjacency matrix :
0 3 1 6 999 999
3 0 5 999 3 999
1 5 0 5 6 4
6 999 5 0 999 2
999 3 6 999 0 6
999 999 4 2 6 0

Edges of the Minimum Spanning Tree ( MST ) are :
1 Edge (1,3) = 1
2 Edge (4,6) = 2
3 Edge (1,2) = 3
4 Edge (2,5) = 3
5 Edge (3,6) = 4

Minimum cost = 13

Process returned 19 (0x13)   execution time : 78.651 s
Press any key to continue.
```