

Experiment No. 10

Aim: Implement Minimum cost spanning trees-Prim's algorithm

Theoretical Background:

It falls under a class of algorithms called greedy algorithms that find the local optimum in the hopes of finding a global optimum.

We start from one vertex and keep adding edges with the lowest weight until we reach our goal.

The steps for implementing Prim's algorithm are as follows:

1. Initialize the minimum spanning tree with a vertex chosen at random.
2. Find all the edges that connect the tree to new vertices, find the minimum and add it to the tree
3. Keep repeating step 2 until we get a minimum spanning tree

Program:

```
#include<stdio.h>

#include<conio.h>

int a,b,u,v,n,i,j,ne=1;

int visited[10]={0},min,mincost=0,cost[10][10];

void main()

{

clrscr();

printf("\n Enter the number of nodes:");

scanf("%d",&n);

printf("\n Enter the 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;

}
```

```
visited[1]=1;

printf("\n");

while(ne<n)

{
for(i=1,min=999;i<=n;i++)
for(j=1;j<=n;j++)
if(cost[i][j]<min)
if(visited[i]!=0)
{
min=cost[i][j];
a=u=i;
b=v=j;
}
if(visited[u]==0 || visited[v]==0)
{
printf("\n Edge %d:(%d %d) cost:%d",ne++,a,b,min);
mincost+=min;
visited[b]=1;
}
cost[a][b]=cost[b][a]=999;
}

printf("\n Minimun cost=%d",mincost);

getch();
}
```

Output:

```
Enter the number of nodes:4

Enter the adjacency matrix:
0      20      10      50
20      0      60      999
10      60      0      40
50      999      40      0

Edge 1:(1 3) cost:10
Edge 2:(1 2) cost:20
Edge 3:(3 4) cost:40
Minimun cost=70
```

Conclusion:

Thus, in this experiment we have studied about implementing Minimum cost spanning trees using Prim's algorithm