**EXPERIMENT – 8**

**AIM**: **Implement prim’s algorithm**

**Input:**

# include<stdio.h>

# include<conio.h>

# define SIZE 20

# define INFINITY 32767

/\* This function finds the minimal spanning tree by Prim's Algorithm \*/

void Prim(int G[][SIZE], int nodes)

{

int select[SIZE], i, j, k;

int min\_dist, v1, v2,total=0;

for (i=0 ; i<nodes ; i++) // Initialize the selected vertices list

select[i] = 0;

printf("\n\n The Minimal Spanning Tree Is :\n");

select[0] = 1;

for (k=1 ; k<nodes ; k++)

{

min\_dist = INFINITY;

for (i=0 ; i<nodes ; i++) // Select an edge such that one vertex is

{ // selected and other is not and the edge

for (j=0 ; j<nodes ; j++) // has the least weight.

{

if (G[i][j] && ((select[i] && !select[j]) || (!select[i] && select[j])))

{

if (G[i][j] < min\_dist)//obtained edge with minimum wt

{

min\_dist = G[i][j];

v1 = i;

v2 = j; //picking up those vertices

}

}

}

}

printf("\n Edge (%d %d )and weight = %d",v1,v2,min\_dist);

select[v1] = select[v2] = 1;

total =total+min\_dist;

}

printf("\n\n\t Total Path Length Is = %d",total);

}

void main()

{

int G[SIZE][SIZE], nodes;

int v1, v2, length, i, j, n;

clrscr();

printf("\n\t Prim'S Algorithm\n");

nodes=4;

n=4;

for (i=0 ; i<nodes ; i++) // Initialize the graph

for (j=0 ; j<nodes ; j++)

G[i][j] = 0;

G[0][1]=G[1][0]=1;

G[0][2]=G[2][0]=5;

G[0][3]=G[3][0]=2;

G[2][3]=G[3][2]=3;

printf("\n Graph is created");

getch();

printf("\n\t");

Prim(G,nodes);

getch();

}