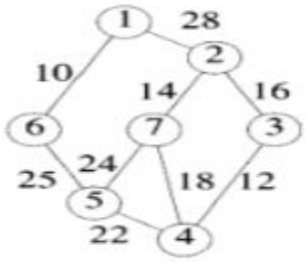
LAB 3 : GREEDY TECHNIQUE

Date : 21.3.18

Submitted By: Abhineet Singh

1. ***Write a program that finds a minimum cost spanning tree of a given undirected graph using Prims Algorithm.Q1. Write a program that finds a minimum cost spanning tree of a given undirected graph using Prims Algorithm.***

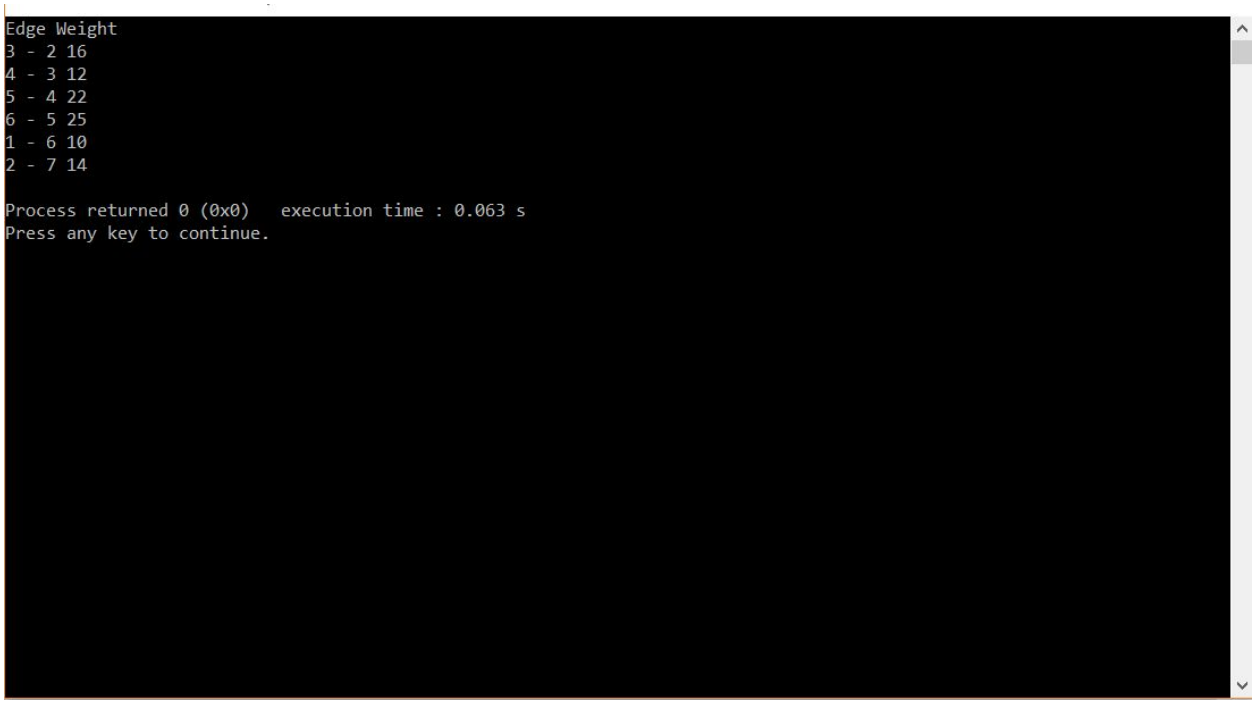



```
#include <stdio.h>
#include <limits.h>
#include <stdbool.h>
#define V 7
int minKey(int key[], bool mstSet[])
{
int min = INT_MAX, min_index;
for (int v = 0; v < V; v++)
if (mstSet[v] == false && key[v] < min)
min = key[v], min_index = v;
return min_index;
}

int printMST(int parent[], int n, int graph[V][V])
{
printf("Edge Weight\n");
for (int i = 1; i < V; i++)
printf("%d - %d %d \n", (parent[i] + 1), (i + 1), graph[i][parent[i]]);
}
void primMST(int graph[V][V])
{
int parent[V];
int key[V];
bool mstSet[V];
```

```
for (int i = 0; i < V; i++)
key[i] = INT_MAX, mstSet[i] = false;
key[0] = 0;
parent[0] = -1;
for (int count = 0; count < V-1; count++)
{
int u = minKey(key, mstSet);
mstSet[u] = true;
for (int v = 0; v < V; v++)
if (graph[u][v] && mstSet[v] == false && graph[u][v] < key[v])
parent[v] = u, key[v] = graph[u][v];
}
printMST(parent, V, graph);
}

int main()
{
int graph[V][V] =   {
                    {0, 28, 0, 0, 0, 10, 0},
                     {28, 0, 16, 0, 0, 0, 14},
                     {0, 16, 0, 12, 0, 0, 0 },
                     {0, 0, 12, 0, 22, 0, 18},
                     {0, 0, 0, 22, 0, 25, 24},
                    {10, 0, 0, 0, 25, 0, 0},
                     {0, 14, 0, 18, 24, 0, 0}
};
primMST(graph);
return 0;
}
```


Edge Weight
3 - 2 16
4 - 3 12
5 - 4 22
6 - 5 25
1 - 6 10
2 - 7 14
Process returned 0 (0x0) execution time : 0.063 s
Press any key to continue.

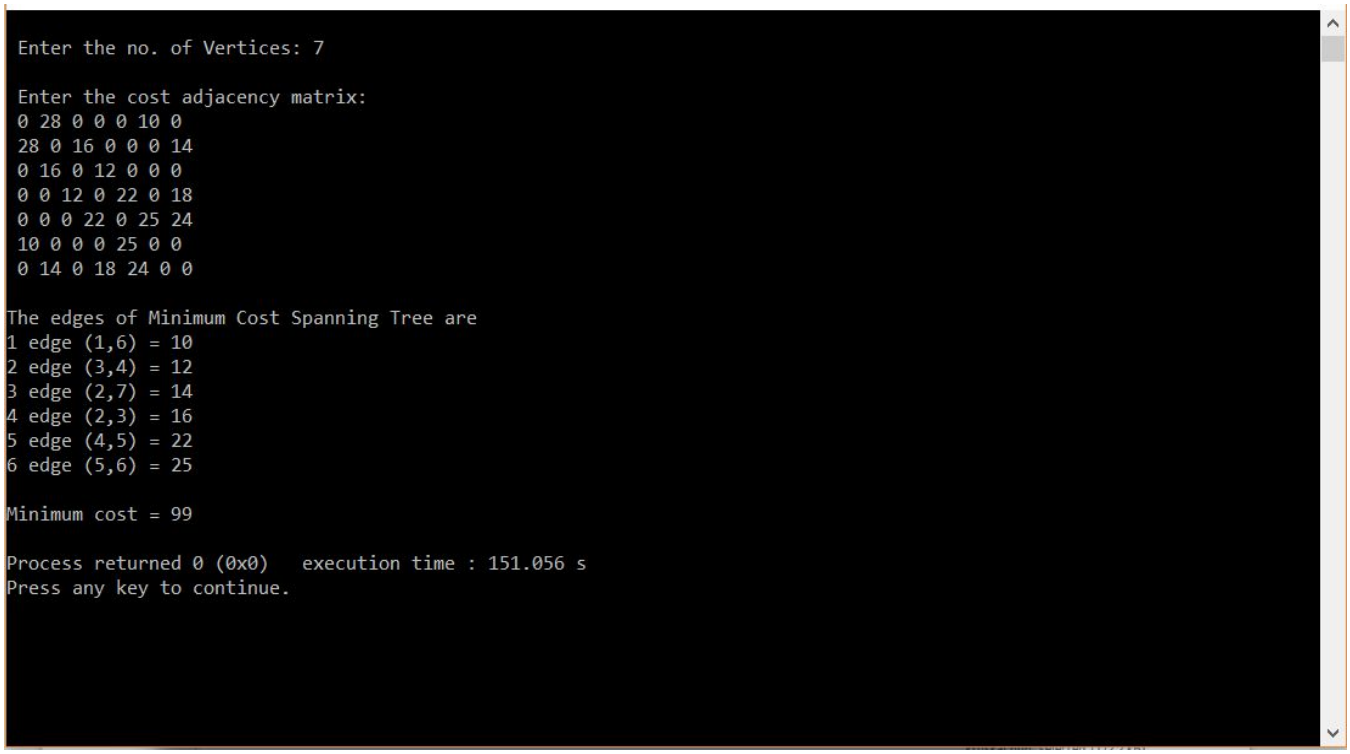
2. ***Write a program that finds a minimum cost spanning tree of a given undirected graph using Kruskal's Algorithm.***

```
#include<stdio.h>
#include<stdlib.h>
int i, j, k, a, b, u, v, n, edge_no = 1;
int min, mincost = 0, cost[9][9], parent[9];
int find(int);
int check(int, int);
int main()
{
printf("\n Enter the no. of Vertices: ");
scanf("%d",&n);
printf("\n Enter the cost 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("\nThe edges of Minimum Cost Spanning Tree are\n");
while(edge_no < 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(check(u, v)){
printf("%d edge (%d,%d) = %d\n", edge_no++, a, b, min);
mincost += min;
}
cost[a][b] = cost[b][a] = 999;
}
```

```
printf("\nMinimum cost = %d\n", mincost);
return 0;
}


int find(int i){
while(parent[i])
i = parent[i];
return i;
}


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


Enter the no. of Vertices: 7
Enter the cost adjacency matrix:
0 28 0 0 0 10 0
28 0 16 0 0 0 14
0 16 0 12 0 0 0
0 0 12 0 22 0 18
0 0 0 22 0 25 24
10 0 0 0 25 0 0
0 14 0 18 24 0 0
The edges of Minimum Cost Spanning Tree are
1 edge (1,6) = 10
2 edge (3,4) = 12
3 edge (2,7) = 14
4 edge (2,3) = 16
5 edge (4,5) = 22
6 edge (5,6) = 25
Minimum cost = 99
Process returned 0 (0x0) execution time : 151.056 s
Press any key to continue.