

Programs:

Prim's:

```
#include <limits.h>
#include <stdbool.h>
#include <stdio.h>
#define V 5

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 graph[V][V])
{
    printf("Edge \tWeight\n");
    for (int i = 1; i < V; i++)
        printf("%d - %d \t%d \n", parent[i], i, 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, graph);
}

int main()
{
    int graph[V][V] = { { 0, 3, 0, 7, 0 },
                        { 3, 0, 2, 4, 9 },
                        { 0, 2, 0, 0, 7 },
                        { 7, 4, 0, 0, 6 },
                        { 0, 9, 7, 6, 0 } };

    primMST(graph);
    return 0; }
```

Output:

```
@pascal-dreams:~/CollegeProject/madf/exp5$ gcc prims.c
@pascal-dreams:~/CollegeProject/madf/exp5$ ./a.out
Edge   Weight
0 - 1   3
1 - 2   2
1 - 3   4
3 - 4   6
```

Kruskal's:

```
#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("\nEnter the no. of vertices:");
    scanf("%d", &n);
    printf("\nEnter 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("The edges of Minimum Cost Spanning Tree 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("\n\tMinimum 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;
}

```

Output:

```

chaitanya@pascal-dreams:~/CollegeProject/madf/exp5$ gcc kruskal.c
chaitanya@pascal-dreams:~/CollegeProject/madf/exp5$ ./a.out

Enter the no. of vertices:3

Enter the cost adjacency matrix:
8
9
6
7
5
6
4
3
2
The edges of Minimum Cost Spanning Tree are
1 edge (3,2) =3
2 edge (3,1) =4

Minimum cost = 7

```