EXERCISE-25

AIM: To write a C program to find the Minimum Spanning Tree (MST) of a connected, weighted graph using Kruskal's Algorithm

Algorithm:

1. Start.

2. Input the number of vertices and edges.

3. Store all edges with their weights.

4. Sort all the edges in increasing order of weights.

5. Use a disjoint set to detect cycles (using Union-Find).

6. Pick the smallest edge. If it doesn't form a cycle, include it in the MST.

7. Repeat until MST has (V - 1) edges.

8. Output the MST and total cost.

9. End.

Program Code:

```c
#include <stdio.h>

#define MAX 30

int parent[MAX];

int find(int i) {

   while (parent[i])

      i = parent[i];

   return i;

}
```

```c
int union_set(int i, int j) {
    if (i != j) {
        parent[j] = i;
        return 1;
    }
    return 0;
}
int main() {
    int u, v, n, i, j, no_of_edges, min, mincost = 0;
    int cost[MAX][MAX], a, b;
    printf("Enter the number of vertices: ");
    scanf("%d", &n);
    printf("Enter the cost adjacency matrix (0 if no edge):\n");
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++) {
            scanf("%d", &cost[i][j]);
            if (cost[i][j] == 0)
                cost[i][j] = 999;
        }
    printf("\nEdges in the Minimum Spanning Tree:\n");
    no_of_edges = 0;
    while (no_of_edges < n - 1) {
        min = 999;
```

```c
    for (i = 0; i < n; i++) {
        for (j = 0; 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 (union_set(u, v)) {
        printf("%d - %d : %d\n", a, b, min);
        mincost += min;
        no_of_edges++;
    }
    cost[a][b] = cost[b][a] = 999; // remove the edge
    }
    printf("Total cost of MST: %d\n", mincost);
    return 0;
}
```

Input and Output:

```
Enter the number of vertices: 4
Enter the cost adjacency matrix (0 if no edge):
0 10 6 5
10 0 0 15
6 0 0 4
5 15 4 0

Edges in the Minimum Spanning Tree:
2 - 3 : 4
0 - 3 : 5
0 - 2 : 6
Total cost of MST: 15
```

Result:

The program successfully computes the Minimum Spanning Tree
(MST) using Kruskal's Algorithm, and displays the selected edges and
the total cost.