**Experiment 25: Kruskal's Algorithm**

Aim:
To write a C program to construct a Minimum Spanning Tree using Kruskal's Algorithm.

Algorithm:

1. Start the program.
2. Sort all edges in increasing order of weight.
3. Initialize disjoint sets for vertices.
4. Pick the smallest edge that does not form a cycle (using union-find).
5. Add the edge to MST.
6. Repeat until V−1 edges are included.
7. Stop.

Code:

```
#include <stdio.h>
#define V 5
#define E 7
struct Edge {
   int u, v, w;
};
int parent[V];
int find(int i) {
   while (i != parent[i]) i = parent[i];
   return i;
}
void unionSet(int u, int v) {
   parent[v] = u;
}
void kruskal(struct Edge edges[], int n) {
   for (int i = 0; i < V; i++) parent[i] = i;
   int count = 0, i = 0;
   printf("Edge : Weight\n");
   while (count < V - 1 && i < n) {
```

```
        int u = find(edges[i].u);
        int v = find(edges[i].v);
        if (u != v) {
            printf("%d - %d : %d\n", edges[i].u, edges[i].v, edges[i].w);
            unionSet(u, v);
            count++;
        }
        i++;
    }
}

int main() {
    struct Edge edges[E] = {
        {0,1,2}, {0,3,6}, {1,2,3}, {1,3,8},
        {1,4,5}, {2,4,7}, {3,4,9}
    };
    kruskal(edges, E);
    return 0;
}
```

Sample Output:

```
Edge : Weight
0 - 1 : 2
0 - 3 : 6
1 - 2 : 3
1 - 4 : 5


=== Code Execution Successful ===
```

Result:
The program successfully finds the MST using Kruskal's Algorithm.