PORT CITY INTERNATIONAL UNIVERSITY

Course Code:

Course Title:

Report Name:

Submitted To:

Name of Lecturer :

Department :

Submitted By:

Name of Student : Mohammed Arsalan Morshed

Program : B.Sc. in CSE

Batch No : CSE-30-D-A2

ID No : CSE-03007824

Date of submission :

Experiment No: 05

Experiment Name: Finding Minimum Spanning Tree using Kruskal's Algorithm

Code:

```
def find(parent, i):
    if parent[i] != i:
        parent[i] = find(parent, parent[i])  # Path compression
    return parent[i]


def union(parent, rank, x, y):
    xroot = find(parent, x)
    yroot = find(parent, y)


    if rank[xroot] < rank[yroot]:
        parent[xroot] = yroot
    elif rank[xroot] > rank[yroot]:
        parent[yroot] = xroot
    else:
        parent[yroot] = xroot
        rank[xroot] += 1


def kruskalMST(V, edges):


    edges.sort(key=lambda edge: edge[2])


    parent = [i for i in range(V)]
    rank = [0] * V
    mst = []
    total_weight = 0
    for src, dest, weight in edges:
        x = find(parent, src)
        y = find(parent, dest)

        if x != y:
            mst.append((src, dest, weight))
            total_weight += weight
            union(parent, rank, x, y)

    print("Minimum Spanning Tree:")
    for src, dest, weight in mst:
        print(f"({src}, {dest}) -> {weight}")
    print(f"Total Minimum Spanning Tree Weight: {total_weight}")

V = 4
edges = [
    (0, 1, 9),
    (1, 2, 11),
    (1, 3, 15),
    (2, 3, 17),
    (2, 3, 4)
]
kruskalMST(V, edges)
```

Output:

```
Minimum Spanning Tree:
(2, 3) -> 4
(0, 1) -> 9
(1, 2) -> 11
Total Minimum Spanning Tree Weight: 24
```