```
class DisjointSet:
    def __init__(self, vertices):
        self.parent = {v: v for v in vertices}
        self.rank = {v: 0 for v in vertices}

    def find(self, v):
        if self.parent[v] != v:
            self.parent[v] = self.find(self.parent[v])
        return self.parent[v]

    def union(self, u, v):
        root1 = self.find(u)
        root2 = self.find(v)

        if root1 != root2:
            if self.rank[root1] > self.rank[root2]:
                self.parent[root2] = root1
            elif self.rank[root1] < self.rank[root2]:
                self.parent[root1] = root2
            else:
                self.parent[root2] = root1
                self.rank[root1] += 1

def kruskal_algorithm(vertices, edges):
    mst = []
    total_weight = 0

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

    for u, v, weight in edges:
        if ds.find(u) != ds.find(v):
            ds.union(u, v)
            mst.append((u, v, weight))
            total_weight += weight

    return mst, total_weight

# Correctly placed vertices and edges outside the function
vertices = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N','O','P','Q','R','S','T']
edges = [
    # Graph1
    ('A','B',1), ('B','C',2), ('C','D',3), ('C','E',5), ('D','F',4),('E','F',6),('F','G',7),

    # Graph2
    ('H','I',6),('H','L',7),('L','M',5),('I','J',9),('J','K',8),('J','L',4),

    # Graph3
    ('N','O',9),('N','P',8),('O','P',7),('P','Q',6),

    # Graph4
    ('S','R',15),('R','T',1),('S','T',14)
]

# Running Kruskal's algorithm
mst, total_weight = kruskal_algorithm(vertices, edges)

print("Minimum Cost Spanning Tree (for all connected components):")
for u, v, weight in mst:
    print(f"{u} - {v}: {weight}")
print(f"\nTotal Weight of Minimum Spanning Trees: {total_weight}")
```

```
Minimum Cost Spanning Tree (for all connected components):
A - B: 1
R - T: 1
B - C: 2
C - D: 3
D - F: 4
J - L: 4
C - E: 5
L - M: 5
H - I: 6
P - Q: 6
F - G: 7
H - L: 7
O - P: 7
```

```
J - K: 8
N - P: 8
S - T: 14

Total Weight of Minimum Spanning Trees: 88
```

Start coding or generate with AI.