**Assignment 12**

**Name: Mudit Mohit**

**Enrolment No.: BT21GCS057**

**Section: B2**

**Question:**  Implement Kruskal’s algorithm to solve the problem of finding a minimum spanning tree. Assume that there are n nodes in the graph and that they have associated costs. Analyze the complexity of the algorithm and calculate the optimal path and its cost value.

Ans.

class DSU:

    def \_\_init\_\_(self, n):

        self.parent = list(range(n))

        self.rank = [0] \* n

    def find(self, x):

        if self.parent[x] != x:

            self.parent[x] = self.find(self.parent[x])

        return self.parent[x]

    def union(self, x, y):

        px, py = self.find(x), self.find(y)

        if px == py:

            return False

        if self.rank[px] < self.rank[py]:

            self.parent[px] = py

        elif self.rank[px] > self.rank[py]:

            self.parent[py] = px

        else:

            self.parent[py] = px

            self.rank[px] += 1

        return True

def kruskal(n, edges):

    dsu = DSU(n)

    edges.sort(key=lambda e: e[2])

    min\_cost = 0

    tree = []

    for u, v, w in edges:

        if dsu.union(u, v):

            tree.append((u, v))

            min\_cost += w

            if len(tree) == n-1:

                break

    return min\_cost, tree

n = 4

edges = [(0, 1, 1), (1, 2, 3), (2, 3, 4), (3, 0, 2)]

min\_cost, tree = kruskal(n, edges)

print("Minimum cost:", min\_cost)

print("Tree edges:", tree)

Output:



Analysis:

The complexity of Kruskal's algorithm is O (E log E), where E is the total number of edges in the graph. This is because the algorithm sorts the edges before processing them, and sorting takes O (E log E) time. After the sort, the algorithm iterates over all the edges, and checking for cycles takes O (log V) time, where V is the total number of nodes in the graph. Since there can be at most E edges and V nodes, the total complexity of the algorithm is O (E log E + E log V). However, since E is always greater than or equal to V-1 in a connected graph, we can simplify the complexity to O (E log E).