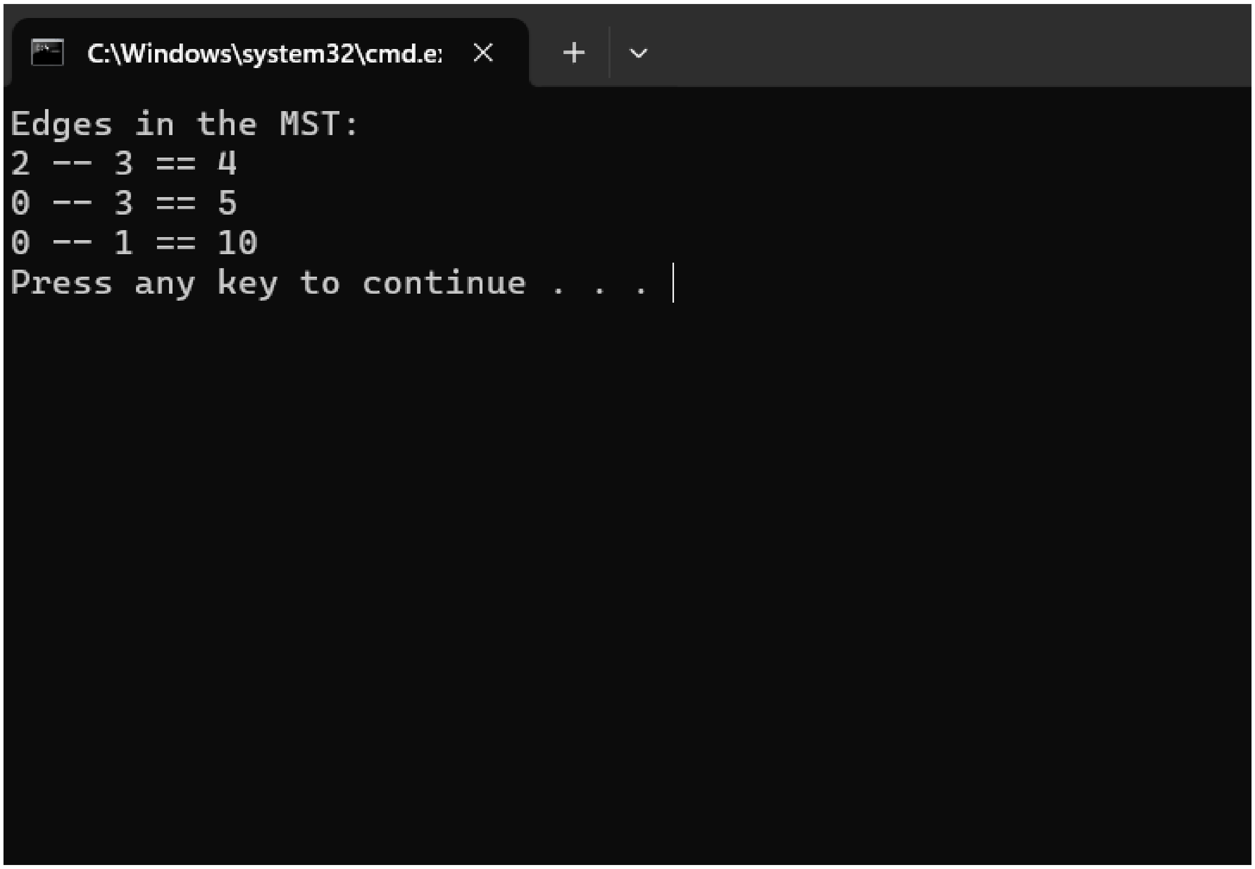
95) Kruskal's Algorithms

CODE:

```
class Graph:
    def __init__(self, vertices):
        self.V = vertices
        self.edges = []
    def add_edge(self, u, v, w):
        self.edges.append((w, u, v))

    def find_parent(self, parent, i):
        if parent[i] != i:
            parent[i] = self.find_parent(parent, parent[i])
        return parent[i]

    def union(self, parent, rank, x, y):
        root_x = self.find_parent(parent, x)
        root_y = self.find_parent(parent, y)

        if rank[root_x] < rank[root_y]:
            parent[root_x] = root_y
        elif rank[root_x] > rank[root_y]:
            parent[root_y] = root_x
        else:
            parent[root_y] = root_x
            rank[root_x] += 1

    def kruskal_mst(self):
        result = []
        self.edges.sort()
        parent = list(range(self.V))
        rank = [0] * self.V

        for w, u, v in self.edges:
            root_u = self.find_parent(parent, u)
            root_v = self.find_parent(parent, v)

            if root_u != root_v:
                result.append((u, v, w))
                self.union(parent, rank, root_u, root_v)

        print("Edges in the MST:")
        for u, v, weight in result:
            print(f"{u} -- {v} == {weight}")

g = Graph(4)
g.add_edge(0, 1, 10)
g.add_edge(0, 2, 6)
g.add_edge(0, 3, 5)
g.add_edge(1, 3, 15)
g.add_edge(2, 3, 4)

g.kruskal_mst()
```

OUTPUT:



```
Edges in the MST:
2 -- 3 == 4
0 -- 3 == 5
0 -- 1 == 10
Press any key to continue . . .
```

TIME COMPLEXITY : O(nlogn)