Kruskal's Algorithm

Code:

In [1]:

```
graph = []
V = 4

def addEdge(u, v, w):
    graph.append([u, v, w])

def find(parent, i):
    if parent[i] != i:
        parent[i] = find(parent, parent[i])
    return parent[i]

def union(parent, rank, x, y):
    if rank[x] < rank[y]:
        parent[x] = y
    elif rank[x] > rank[y]:
        parent[y] = x
    else:
        parent[y] = x
        rank[x] += 1

def KruskalMST():
    result = []
    parent = []
    rank = []
    i = 0
    e = 0
    graph.sort(key=lambda item: item[2])
    for node in range(V):
        parent.append(node)
        rank.append(0)

    while e < V - 1:
        u, v, w = graph[i]
        i = i + 1
        x = find(parent, u)
        y = find(parent, v)
        if x != y:
            e = e + 1
            result.append([u, v, w])
            union(parent, rank, x, y)

    minimumCost = 0
    print("Edge : Weight")
    for u, v, weight in result:
        minimumCost += weight
        print (" " + str(u) + "-" + str(v) + " : " + str(weight))
    print("\nMinimum Spanning Tree: ", minimumCost)

addEdge(0, 1, 10)
addEdge(0, 2, 6)
addEdge(0, 3, 5)
addEdge(1, 3, 15)
addEdge(2, 3, 4)

KruskalMST()
```

```
#Output
```

```
Edge : Weight
 2-3 : 4
 0-3 : 5
 0-1 : 10

Minimum Spanning Tree:  19
```