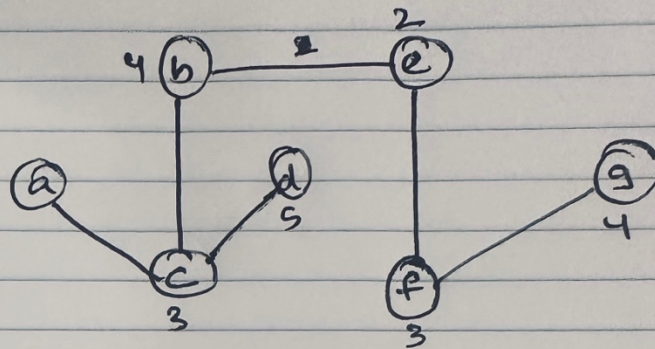


## Week5-Q2

Krushal's method:

## Week 5 - Q2

weight	link
2	b-e
3	a-c
3	e-f
4	<del>b-c</del> c-b
4	f-g
5	a-b
5	c-d
5	e-g
6	b-d
6	c-f
6	d-e
6	d-f



Total weight

$$= 3 + 4 + 5 + 2 + 3 + 4$$

$$= 21$$

```

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

def union(parent, rank, x, y):
    x_root = find(parent, x)
    y_root = find(parent, y)

    if rank[x_root] < rank[y_root]:
        parent[x_root] = y_root
    elif rank[x_root] > rank[y_root]:
        parent[y_root] = x_root
    else:
        parent[y_root] = x_root
        rank[x_root] += 1

def minimumCost(n, connections):
    # Sort the connections based on their costs
    connections.sort(key=lambda x: x[2])

    # Initialize parent and rank arrays for union-find
    parent = [i for i in range(n + 1)]
    rank = [0] * (n + 1)

    min_cost = 0
    num_edges = 0

    for u, v, cost in connections:
        u_root = find(parent, u)
        v_root = find(parent, v)

        if u_root != v_root:
            union(parent, rank, u_root, v_root)
            min_cost += cost
            num_edges += 1

        if num_edges == n - 1:
            break

    # Check if all cities are connected
    if num_edges == n - 1:
        return min_cost
    else:

```

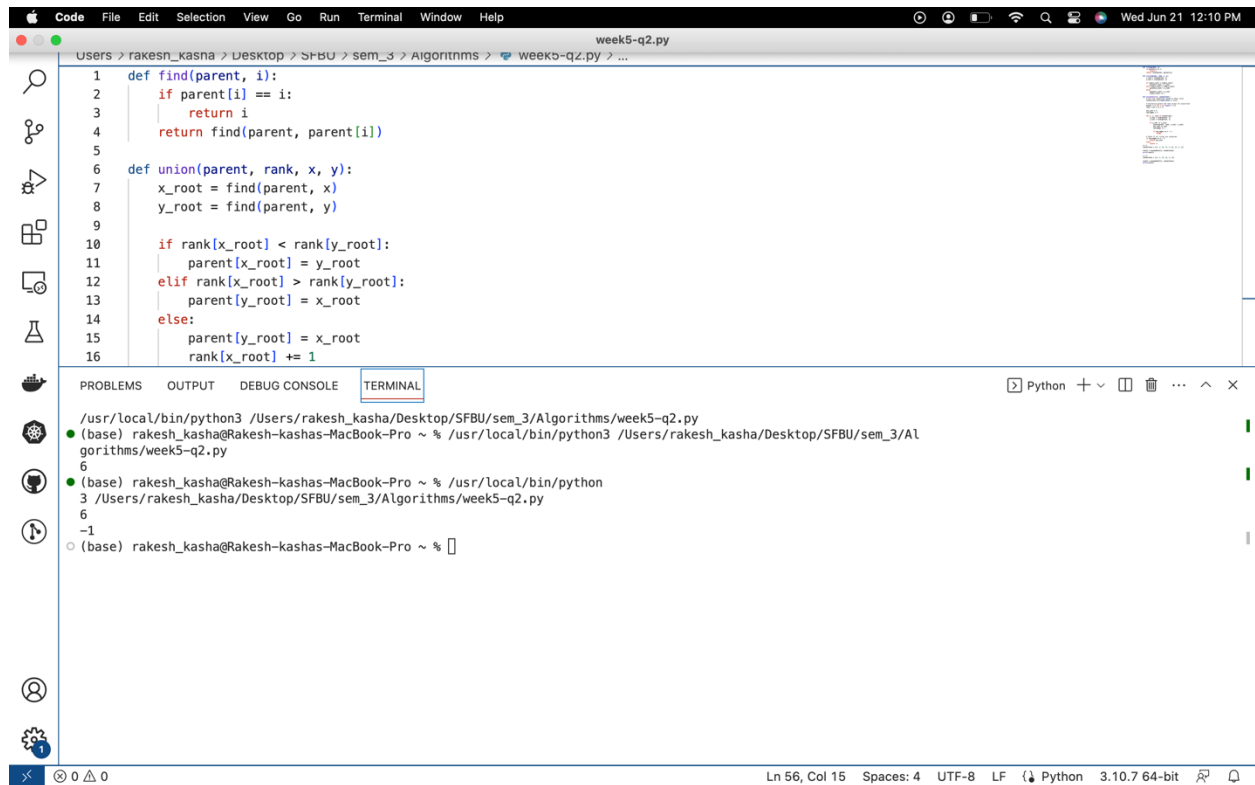
```
return -1
```

```
n = 3
```

```
connections = [[1, 2, 5], [1, 3, 6], [2, 3, 1]]
```

```
result = minimumCost(n, connections)
```

```
print(result)
```



The screenshot shows a VS Code editor window with a file named `week5-q2.py` open. The file contains a Union-Find data structure implementation. The code is as follows:

```
1 def find(parent, i):
2     if parent[i] == i:
3         return i
4     return find(parent, parent[i])
5
6 def union(parent, rank, x, y):
7     x_root = find(parent, x)
8     y_root = find(parent, y)
9
10    if rank[x_root] < rank[y_root]:
11        parent[x_root] = y_root
12    elif rank[x_root] > rank[y_root]:
13        parent[y_root] = x_root
14    else:
15        parent[y_root] = x_root
16        rank[x_root] += 1
```

The terminal output shows the execution of the script:

```
/usr/local/bin/python3 /Users/rakesh_kasha/Desktop/SFBU/sem_3/Algorithms/week5-q2.py
(base) rakesh_kasha@Rakesh-kashas-MacBook-Pro ~ % /usr/local/bin/python3 /Users/rakesh_kasha/Desktop/SFBU/sem_3/Algorithms/week5-q2.py
6
(base) rakesh_kasha@Rakesh-kashas-MacBook-Pro ~ % /usr/local/bin/python3
3 /Users/rakesh_kasha/Desktop/SFBU/sem_3/Algorithms/week5-q2.py
6
-1
(base) rakesh_kasha@Rakesh-kashas-MacBook-Pro ~ %
```

The status bar at the bottom indicates the current position is Line 56, Column 15, with 4 spaces, using UTF-8 encoding, LF line endings, Python 3.10.7 64-bit.

CodeFileEditSelectionViewGoRunTerminalWindowHelp

Users > rakesh\_kasha > Desktop > SFBU > sem\_3 > Algorithms > week5-q2.py > ...

43return min\_cost

44else:

45return -1

46n = 3

47connections = [[1, 2, 5], [1, 3, 6], [2, 3, 1]]

48

49result = minimumCost(n, connections)

50print(result)

51

52n = 4

53connections = [[1, 2, 3], [3, 4, 4]]

54

55result = minimumCost(n, connections)

56print(result)

PROBLEMSOUTPUTDEBUG CONSOLETERMINAL

/usr/local/bin/python3 /Users/rakesh\_kasha/Desktop/SFBU/sem\_3/Algorithms/week5-q2.py

● (base) rakesh\_kasha@Rakesh-kashas-MacBook-Pro ~ % /usr/local/bin/python3 /Users/rakesh\_kasha/Desktop/SFBU/sem\_3/Algorithms/week5-q2.py

6

● (base) rakesh\_kasha@Rakesh-kashas-MacBook-Pro ~ % /usr/local/bin/python

3 /Users/rakesh\_kasha/Desktop/SFBU/sem\_3/Algorithms/week5-q2.py

6

-1

○ (base) rakesh\_kasha@Rakesh-kashas-MacBook-Pro ~ %

Ln 56, Col 15Spaces: 4UTF-8LFPython 3.10.7 64-bit