
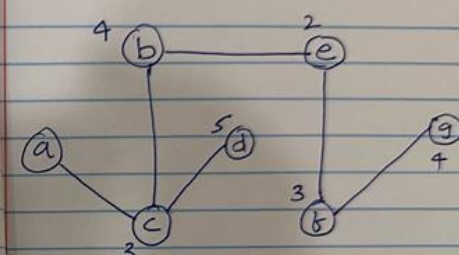


Hw5q2

Kruskal approach

weight	path
2	b-e
3	a-c
3	e-f
4	b-b
4	f-g
5	a-b cannot be joined
5	c-d
5	e-g cannot be joined
6	b-d cannot be joined
6	d-e cannot be joined
6	d-f cannot be joined
6	c-f cannot be joined





Total weight = $3+4+5+2+3+4$
 $= 21$

Chatgpt python code

```
def minimumCost(n, connections):
    def find(parent, city):
        if parent[city] != city:
            parent[city] = find(parent, parent[city]) # Path compression
        return parent[city]

    def union(parent, rank, city1, city2):
        root1 = find(parent, city1)
        root2 = find(parent, city2)
        if rank[root1] < rank[root2]:
            parent[root1] = root2
        elif rank[root1] > rank[root2]:
            parent[root2] = root1
        else:
            parent[root2] = root1
            rank[root1] += 1

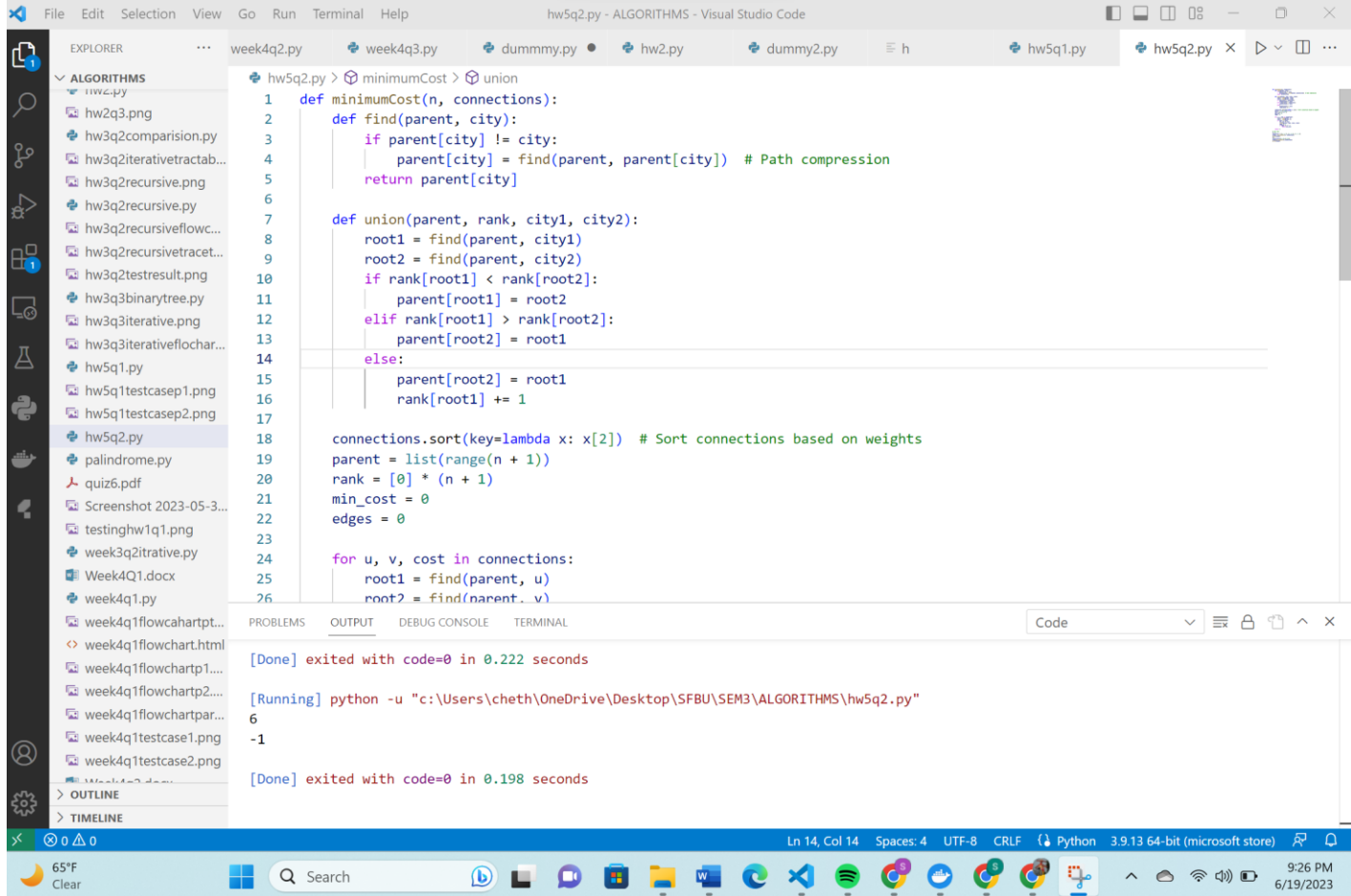
    connections.sort(key=lambda x: x[2]) # Sort connections based on weights
    parent = list(range(n + 1))
    rank = [0] * (n + 1)
    min_cost = 0
    edges = 0

    for u, v, cost in connections:
        root1 = find(parent, u)
        root2 = find(parent, v)
        if root1 != root2:
            min_cost += cost
            union(parent, rank, root1, root2)
            edges += 1
            if edges == n - 1:
                return min_cost

    return -1

# Test case
n = 3
connections = [[1, 2, 5], [1, 3, 6], [2, 3, 1]]
output = minimumCost(n, connections)
print(output)
```

testcases



The screenshot displays the Visual Studio Code interface with the following components:

- Explorer Sidebar:** Lists files under the 'ALGORITHMS' folder, including `hw2q3.png`, `hw3q2comparison.py`, `hw3q2iterativetractab...`, `hw3q2recursive.png`, `hw3q2recursive.py`, `hw3q2recursiveflowc...`, `hw3q2recursivevtracet...`, `hw3q2testresult.png`, `hw3q3binarytree.py`, `hw3q3iterative.png`, `hw3q3iterativeflochar...`, `hw5q1.py`, `hw5q1testcasep1.png`, `hw5q1testcasep2.png`, `hw5q2.py` (selected), `palindrome.py`, `quiz6.pdf`, `Screenshot 2023-05-3...`, `testinghw1q1.png`, `week3q2itrative.py`, `Week4Q1.docx`, `week4q1.py`, `week4q1flowchartpt...`, `week4q1flowchart.html`, `week4q1flowchartp1....`, `week4q1flowchartp2....`, `week4q1flowchartpar...`, `week4q1testcase1.png`, and `week4q1testcase2.png`. It also shows 'OUTLINE' and 'TIMELINE' sections.
- Editor Window:** Displays the code for `hw5q2.py` with the following content:

```
1 def minimumCost(n, connections):
2     def find(parent, city):
3         if parent[city] != city:
4             parent[city] = find(parent, parent[city]) # Path compression
5         return parent[city]
6
7     def union(parent, rank, city1, city2):
8         root1 = find(parent, city1)
9         root2 = find(parent, city2)
10        if rank[root1] < rank[root2]:
11            parent[root1] = root2
12        elif rank[root1] > rank[root2]:
13            parent[root2] = root1
14        else:
15            parent[root2] = root1
16            rank[root1] += 1
17
18    connections.sort(key=lambda x: x[2]) # Sort connections based on weights
19    parent = list(range(n + 1))
20    rank = [0] * (n + 1)
21    min_cost = 0
22    edges = 0
23
24    for u, v, cost in connections:
25        root1 = find(parent, u)
26        root2 = find(parent, v)
```
- Output Window:** Shows the execution results:

```
[Done] exited with code=0 in 0.222 seconds
[Running] python -u "c:\Users\cheth\OneDrive\Desktop\SFBU\SEM3\ALGORITHMS\hw5q2.py"
6
-1
[Done] exited with code=0 in 0.198 seconds
```
- Status Bar:** Indicates 'Ln 14, Col 14', 'Spaces: 4', 'UTF-8', 'CRLF', 'Python', and '3.9.13 64-bit (microsoft store)'. The system tray shows a temperature of 65°F, a search bar, and the date/time '9:26 PM 6/19/2023'.

Visual Studio Code interface showing a Python file named `hw5q2.py` in the `ALGORITHMS` folder. The code implements a Union-Find algorithm for finding the minimum cost of connecting nodes.

```
24 for u, v, cost in connections:
25     root1 = find(parent, u)
26     root2 = find(parent, v)
27     if root1 != root2:
28         min_cost += cost
29         union(parent, rank, root1, root2)
30         edges += 1
31         if edges == n - 1:
32             return min_cost
33
34 return -1
35
36 # Test case
37 n = 3
38 connections = [[1, 2, 5], [1, 3, 6], [2, 3, 1]]
39 output = minimumCost(n, connections)
40 print(output)
41 n=4
42 connections=[[1,2,3],[3,4,4]]
43 output=minimumCost(n,connections)
44 print(output)
```

The terminal output shows the execution of the script:

```
[Done] exited with code=0 in 0.222 seconds
[Running] python -u "c:\Users\cheth\OneDrive\Desktop\SFBU\SEM3\ALGORITHMS\hw5q2.py"
6
-1
[Done] exited with code=0 in 0.198 seconds
```

The status bar at the bottom indicates the file is at Line 14, Column 14, using UTF-8 encoding and CRLF line endings. The Python version is 3.9.13 64-bit (microsoft store).

System tray information: 65°F Clear, 9:27 PM, 6/19/2023.