

DAA LAB : 19 NOV 2024

ADITYA MADWAL

2200290120011

The screenshot displays a coding platform interface with a dark theme. The top navigation bar includes links for Courses, Tutorials, Jobs, Practice, and Contests. The main header shows the platform's logo and a user profile icon. The left sidebar contains a menu with options like Problem, Editorial, Submissions, and Comments. The central area is divided into two main sections: the left section shows the 'Output Window' with 'Compilation Results' and 'Custom Input' tabs, and the right section shows the code editor. The 'Problem Solved Successfully' message is prominently displayed in the left section, along with statistics: 71/71 Test Cases Passed, 2/4 Attempts, and 50% Accuracy. The code editor on the right shows a Python3 solution for a Minimum Spanning Tree problem, featuring a 'spanningTree' function and a 'find' function. The bottom of the interface has a 'Solve Next' button and a 'Submit' button.

Courses ▾ Tutorials ▾ Jobs ▾ Practice ▾ Contests ▾

Python3 Average Time: 25m Start Timer

Output Window

Compilation Results Custom Input Y.O.G.I. (AI Bot)

Problem Solved Successfully [Suggest Feedback](#)

Test Cases Passed
71 / 71

Attempts : Correct / Total
2 / 4

Accuracy : 50%

Time Taken
0.37

You get marks only for the first correct submission if you solve the problem without viewing the full solution.

Solve Next

```
1 #User function Template for python3
2
3 from typing import List
4 class Solution:
5
6     #Function to find sum of weights of edges of the Minimum Spanning
7     #code here
8     def find(p, x):
9         while p[x] != x:
10             x = p[x]
11         return x
12
13
14
15     def union(p, rank, x, y):
16         root_x = find(p, x)
17         root_y = find(p, y)
18         if root_x != root_y:
19             if rank[root_x] > rank[root_y]:
20                 p[root_y] = root_x
21             elif rank[root_x] < rank[root_y]:
22                 p[root_x] = root_y
23             else:
24                 p[root_y] = root_x
25                 rank[root_x] += 1
26             return True
27         return False
28
29 n = V
30 k = 1
31
32 p = [i for i in range(n + 1)]
```

Custom Input Compile & Run Submit