GRAPHS...

video-75

"let's make it easy too"

Medium

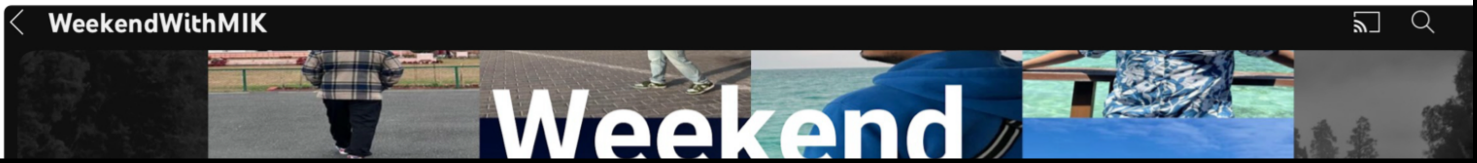
Leetcode-3372

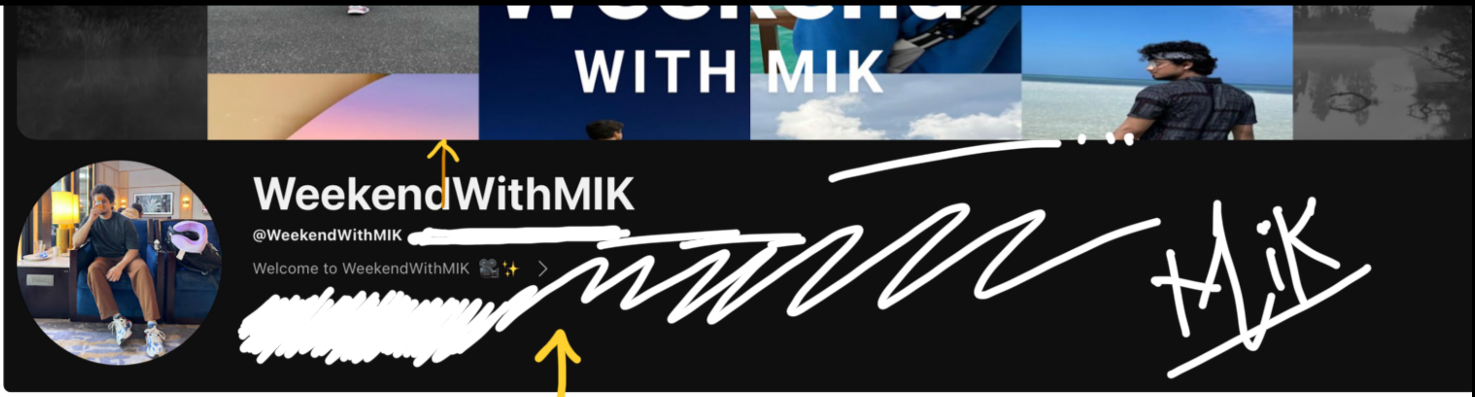
If you have tried my "Graph Concepts & Qns" playlist, these Qns, will seem very easy. Do try it once :)

Facebook
Instagram → codestorywithMIK

(Twitter) → CSwithMIK

codestorywithMIK →





WeekendWithMIK

@WeekendWithMIK

Welcome to WeekendWithMIK 🎥✨ >

[redacted]

MIK

Try this channel to see 'Life behind the Scenes' + 'Tech News'

Motivation :-

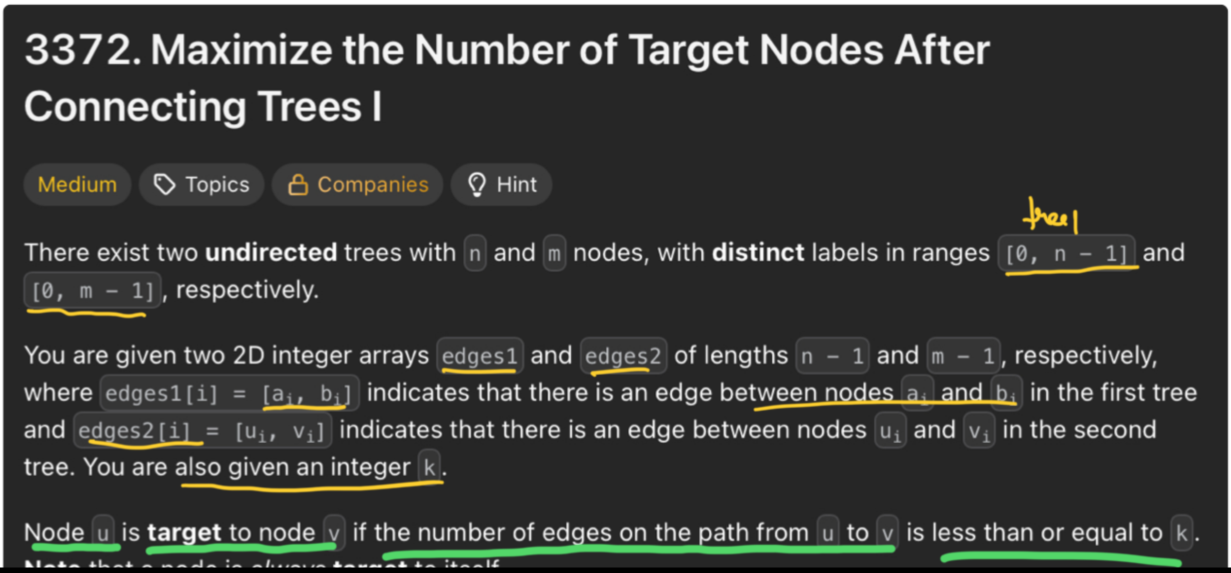
Why are you scared & worried?

Instead, convert that energy to hard work and more & more rigorous preparation.

Your fear will be gone, you will feel confident and things will change for you.

(Talking from experience)

MIK...



3372. Maximize the Number of Target Nodes After Connecting Trees I

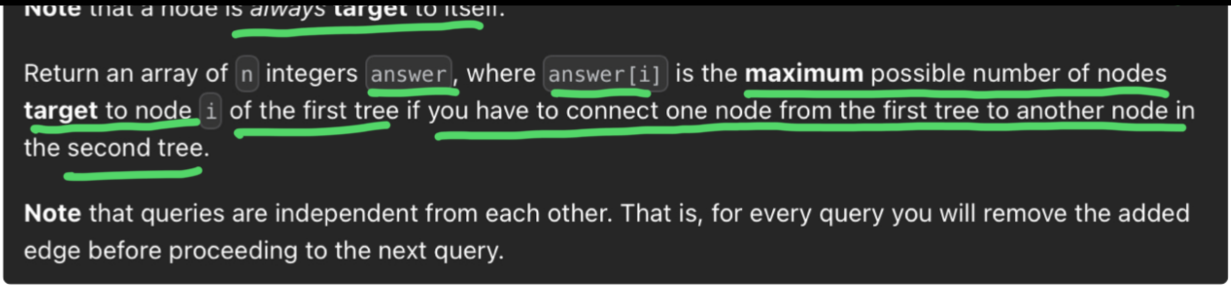
Medium | Topics | Companies | Hint

There exist two **undirected** trees with n and m nodes, with **distinct** labels in ranges [0, n - 1] and [0, m - 1], respectively.

tree1

You are given two 2D integer arrays edges1 and edges2 of lengths n - 1 and m - 1, respectively, where edges1[i] = [a_i, b_i] indicates that there is an edge between nodes a_i and b_i in the first tree and edges2[i] = [u_i, v_i] indicates that there is an edge between nodes u_i and v_i in the second tree. You are also given an integer k.

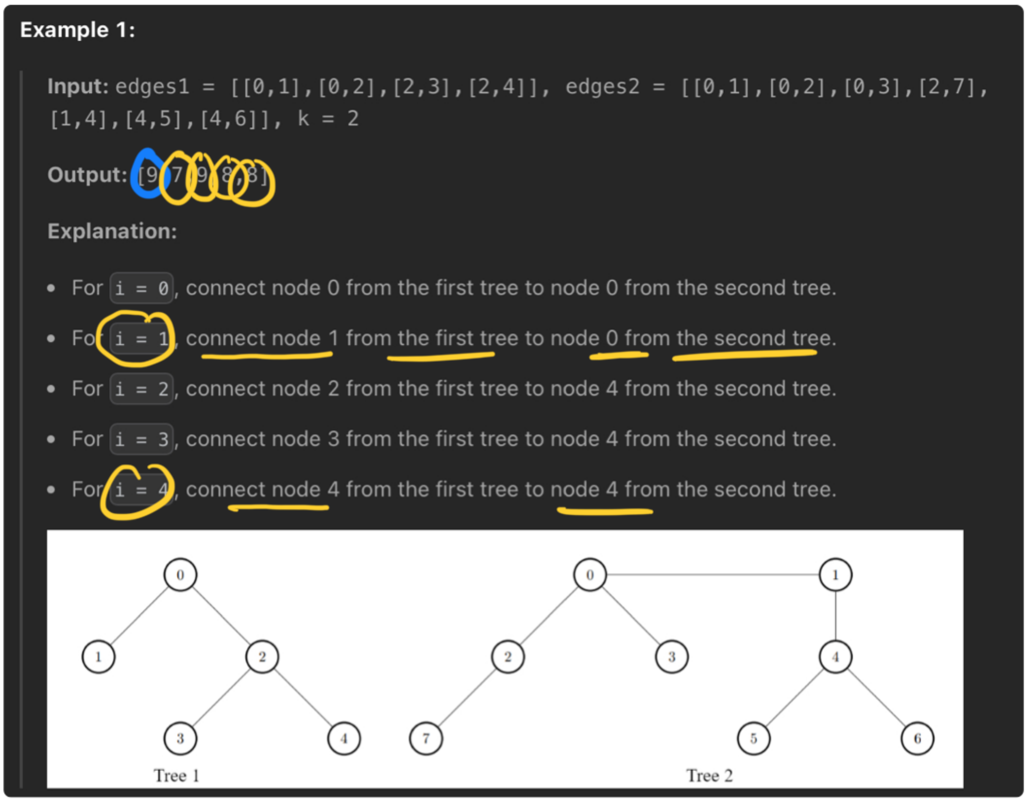
Node u is **target** to node v if the number of edges on the path from u to v is less than or equal to k. Note that a node is *always* **target** to itself.



Note that a node is *always* **target** to itself.

Return an array of `n` integers `answer`, where `answer[i]` is the **maximum** possible number of nodes **target** to node `i` of the first tree if you have to connect one node from the first tree to another node in the second tree.

Note that queries are independent from each other. That is, for every query you will remove the added edge before proceeding to the next query.



Example 1:

Input: edges1 = [[0,1],[0,2],[2,3],[2,4]], edges2 = [[0,1],[0,2],[0,3],[2,7],[1,4],[4,5],[4,6]], k = 2

Output: [9,7,9,8,8]

Explanation:

- For `i = 0`, connect node 0 from the first tree to node 0 from the second tree.
- For `i = 1`, connect node 1 from the first tree to node 0 from the second tree.
- For `i = 2`, connect node 2 from the first tree to node 4 from the second tree.
- For `i = 3`, connect node 3 from the first tree to node 4 from the second tree.
- For `i = 4`, connect node 4 from the first tree to node 4 from the second tree.

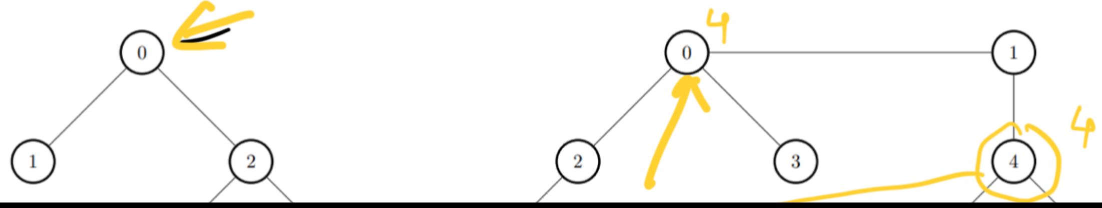
K=2

0	1	2	3	4
9	7	9	8	8

0 1 2 3 4 0 1 2 3

Thought Process

K=2



2

i

result =

0	1	2	3	4
~~5~~	~~3~~	~~5~~	~~4~~	~~4~~
9	7	9	8	8

result2 =

0	1	2	3	4	5	6	7
4	3	3	2	4	2	2	2

, (K-1)

result[i] = DFS(Tree1, i) ;

Story Points:-

1. DFS / BFS precalculate Tree2 target nodes from each node within (K-1) dist.

2. Find max target nodes count from step 1. → maxTargetCount

3. Find target nodes count for each node within tree 1 (K distance)

4

```
result[i] += maxTarget Count;
```