Problem 2581: Count Number of Possible Root Nodes

Problem Information

Difficulty: Hard
Acceptance Rate: 47.50%
Paid Only: No
Tags: Array, Hash Table, Dynamic Programming, Tree, Depth-First Search

Problem Description

Alice has an undirected tree with `n` nodes labeled from `0` to `n - 1`. The tree is represented as a 2D integer array `edges` of length `n - 1` where `edges[i] = [ai, bi]` indicates that there is an edge between nodes `ai` and `bi` in the tree.

Alice wants Bob to find the root of the tree. She allows Bob to make several **guesses** about her tree. In one guess, he does the following:

* Chooses two **distinct** integers `u` and `v` such that there exists an edge `[u, v]` in the tree. * He tells Alice that `u` is the **parent** of `v` in the tree.

Bob's guesses are represented by a 2D integer array `guesses` where `guesses[j] = [uj, vj]` indicates Bob guessed `uj` to be the parent of `vj`.

Alice being lazy, does not reply to each of Bob's guesses, but just says that **at least** `k` of his guesses are `true`.

Given the 2D integer arrays `edges`, `guesses` and the integer `k`, return _the**number of possible nodes** that can be the root of Alice's tree_. If there is no such tree, return `0`.

Example 1:

![](https://assets.leetcode.com/uploads/2022/12/19/ex-1.png)

Input: edges = [[0,1],[1,2],[1,3],[4,2]], guesses = [[1,3],[0,1],[1,0],[2,4]], k = 3 **Output:** 3 **Explanation:** Root = 0, correct guesses = [1,3], [0,1], [2,4] Root = 1, correct guesses =

[1,3], [1,0], [2,4] Root = 2, correct guesses = [1,3], [1,0], [2,4] Root = 3, correct guesses = [1,0], [2,4] Root = 4, correct guesses = [1,3], [1,0] Considering 0, 1, or 2 as root node leads to 3 correct guesses.

Example 2:

![](https://assets.leetcode.com/uploads/2022/12/19/ex-2.png)

Input: edges = [[0,1],[1,2],[2,3],[3,4]], guesses = [[1,0],[3,4],[2,1],[3,2]], k = 1 **Output:** 5 **Explanation:** Root = 0, correct guesses = [3,4] Root = 1, correct guesses = [1,0], [3,4] Root = 2, correct guesses = [1,0], [2,1], [3,4] Root = 3, correct guesses = [1,0], [2,1], [3,2], [3,4] Root = 4, correct guesses = [1,0], [2,1], [3,2] Considering any node as root will give at least 1 correct guess.

Constraints:

* `edges.length == n - 1` * `2 <= n <= 105` * `1 <= guesses.length <= 105` * `0 <= ai, bi, uj, vj <= n - 1` * `ai != bi` * `uj != vj` * `edges` represents a valid tree. * `guesses[j]` is an edge of the tree. * `guesses` is unique. * `0 <= k <= guesses.length`

Code Snippets

C++:

```
class Solution {
public:
int rootCount(vector<vector<int>>& edges, vector<vector<int>>& guesses, int
k) {

}
};
```

Java:

```
class Solution {
public int rootCount(int[][] edges, int[][] guesses, int k) {

}
}
```

Python3:

```
class Solution:
def rootCount(self, edges: List[List[int]], guesses: List[List[int]], k: int)
-> int:
```