Problem 1042: Flower Planting With No Adjacent

Problem Information

Difficulty: Medium
Acceptance Rate: 53.09%
Paid Only: No
Tags: Depth-First Search, Breadth-First Search, Graph

Problem Description

You have `n` gardens, labeled from `1` to `n`, and an array `paths` where `paths[i] = [xi, yi]` describes a bidirectional path between garden `xi` to garden `yi`. In each garden, you want to plant one of 4 types of flowers.

All gardens have **at most 3** paths coming into or leaving it.

Your task is to choose a flower type for each garden such that, for any two gardens connected by a path, they have different types of flowers.

Return _**any** such a choice as an array _`answer` _, where_`answer[i]`_is the type of flower planted in the_`(i+1)th` _garden. The flower types are denoted_`1` _,_`2` _,_`3` _, or_`4` _. It is guaranteed an answer exists._

Example 1:

Input: n = 3, paths = [[1,2],[2,3],[3,1]] **Output:** [1,2,3] **Explanation:** Gardens 1 and 2 have different types. Gardens 2 and 3 have different types. Gardens 3 and 1 have different types. Hence, [1,2,3] is a valid answer. Other valid answers include [1,2,4], [1,4,2], and [3,2,1].

Example 2:

Input: n = 4, paths = [[1,2],[3,4]] **Output:** [1,2,1,2]

Example 3:

Input: n = 4, paths = [[1,2],[2,3],[3,4],[4,1],[1,3],[2,4]] **Output:** [1,2,3,4]

Constraints:

* `1 <= n <= 104` * `0 <= paths.length <= 2 * 104` * `paths[i].length == 2` * `1 <= xi, yi <= n` * `xi != yi` * Every garden has **at most 3** paths coming into or leaving it.

Code Snippets

C++:

```
class Solution {
public:
vector<int> gardenNoAdj(int n, vector<vector<int>>& paths) {

}
};
```

Java:

```
class Solution {
public int[] gardenNoAdj(int n, int[][] paths) {

}
}
```

Python3:

```
class Solution:
def gardenNoAdj(self, n: int, paths: List[List[int]]) -> List[int]:
```