Problem 886: Possible Bipartition

Problem Information

Difficulty: Medium
Acceptance Rate: 52.07%
Paid Only: No
Tags: Depth-First Search, Breadth-First Search, Union Find, Graph

Problem Description

We want to split a group of `n` people (labeled from `1` to `n`) into two groups of **any size**. Each person may dislike some other people, and they should not go into the same group.

Given the integer `n` and the array `dislikes` where `dislikes[i] = [ai, bi]` indicates that the person labeled `ai` does not like the person labeled `bi`, return `true` _if it is possible to split everyone into two groups in this way_.

Example 1:

Input: n = 4, dislikes = [[1,2],[1,3],[2,4]] **Output:** true **Explanation:** The first group has [1,4], and the second group has [2,3].

Example 2:

Input: n = 3, dislikes = [[1,2],[1,3],[2,3]] **Output:** false **Explanation:** We need at least 3 groups to divide them. We cannot put them in two groups.

Constraints:

* `1 <= n <= 2000` * `0 <= dislikes.length <= 104` * `dislikes[i].length == 2` * `1 <= ai < bi <= n` * All the pairs of `dislikes` are **unique**.

Code Snippets

C++:

```
class Solution {
public:
bool possibleBipartition(int n, vector<vector<int>>& dislikes) {

}
};
```

Java:

```
class Solution {
public boolean possibleBipartition(int n, int[][] dislikes) {

}
}
```

Python3:

```
class Solution:
def possibleBipartition(self, n: int, dislikes: List[List[int]]) -> bool:
```