Problem 997: Find the Town Judge

Problem Information

Difficulty: Easy
Acceptance Rate: 50.32%
Paid Only: No
Tags: Array, Hash Table, Graph

Problem Description

In a town, there are `n` people labeled from `1` to `n`. There is a rumor that one of these people is secretly the town judge.

If the town judge exists, then:

1. The town judge trusts nobody. 2. Everybody (except for the town judge) trusts the town judge. 3. There is exactly one person that satisfies properties **1** and **2**.

You are given an array `trust` where `trust[i] = [ai, bi]` representing that the person labeled `ai` trusts the person labeled `bi`. If a trust relationship does not exist in `trust` array, then such a trust relationship does not exist.

Return _the label of the town judge if the town judge exists and can be identified, or return_`-1` _otherwise_.

Example 1:

Input: n = 2, trust = [[1,2]] **Output:** 2

Example 2:

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

Example 3:

Input: n = 3, trust = [[1,3],[2,3],[3,1]] **Output:** -1

Constraints:

* `1 <= n <= 1000` * `0 <= trust.length <= 104` * `trust[i].length == 2` * All the pairs of `trust` are **unique**. * `ai != bi` * `1 <= ai, bi <= n`

Code Snippets

C++:

```
class Solution {
public:
int findJudge(int n, vector<vector<int>>& trust) {

}
};
```

Java:

```
class Solution {
public int findJudge(int n, int[][] trust) {

}
}
```

Python3:

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