

Jewels and Stones

Ransom Note

Number Complement

First Unique Character in a ...

Majority Element

Cousins in Binary Tree

Week 2: May 8th–May 14th
Problems appear at midnight, P.

Check If It Is a Straight Line

Valid Perfect Square

Find the Town Judge

Flood Fill

Single Element in a Sorted ...

< Previous Next >

Go to Discuss



Find the Town Judge

Solution

In a town, there are N people labelled 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 `trust`, an array of pairs `trust[i] = [a, b]` representing that the person labelled `a` trusts the person labelled `b`.

If the town judge exists and can be identified, return the label of the town judge. Otherwise, return `-1`.

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`

Example 4:

Input: $N = 3$, `trust = [[1,2],[2,3]]`
Output: `-1`

Example 5:

Input: $N = 4$, `trust = [[1,3],[1,4],[2,3],[2,4],[4,3]]`
Output: `3`

Note:

1. $1 \leq N \leq 1000$
2. `trust.length` ≤ 10000
3. `trust[i]` are all different
4. `trust[i][0] != trust[i][1]`
5. $1 \leq \text{trust}[i][0], \text{trust}[i][1] \leq N$

Java



```
1 class Solution {
2     public int findJudge(int N, int[][] trust) {
3         if(trust.length==0) return (N==1)?1:-1;
4         int[] arr=new int[N+1];
5         for(int[] t:trust){
6             arr[t[0]]--;
7             arr[t[1]]++;
8         }
9         for(int i=0;i<arr.length;i++){
10             if(arr[i]==N-1)
11                 return i;
12         }
13         return -1;
14     }
15 }
```

☐ Custom Testcase ([Contribute](#) ⓘ)



Run Code

Submit