[LeetCode](https://leetcode.com/problems/find-the-town-judge/description/?envType=daily-question&envId=2024-02-22)

<https://github.com/AdityaKonda6/-50DaysOfCoding>

<https://leetcode.com/problems/find-the-town-judge/description/?envType=daily-question&envId=2024-02-22>

<https://www.linkedin.com/in/aditya-adi-konda/>

 Day 3 of [#50dayscodingchallenge](https://www.linkedin.com/feed/hashtag/?keywords=50dayscodingchallenge&highlightedUpdateUrns=urn:li:activity:7166316239483461633):  
[#leetcode](https://www.linkedin.com/feed/hashtag/?keywords=leetcode&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcodechallenge](https://www.linkedin.com/feed/hashtag/?keywords=leetcodechallenge&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcodestreak](https://www.linkedin.com/feed/hashtag/?keywords=leetcodestreak&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcode2024](https://www.linkedin.com/feed/hashtag/?keywords=leetcode2024&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcode50day](https://www.linkedin.com/feed/hashtag/?keywords=leetcode50day&highlightedUpdateUrns=urn:li:activity:7166316239483461633)  
   
Just kicked off my coding journey with a fascinating problem - "Successfully solved LeetCode Problem “997. Find the Town Judge” !”  
   
✨ Task:   
Rumors are swirling in our town!   
1️⃣ There's exactly one person satisfying both conditions.  
Given an array of trust relationships, can you identify the town judge?   
  
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  
  
Unravel the mystery using your coding skills!  [#CodingChallenge](https://www.linkedin.com/feed/hashtag/?keywords=codingchallenge&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#TownJudge](https://www.linkedin.com/feed/hashtag/?keywords=townjudge&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#Algorithm](https://www.linkedin.com/feed/hashtag/?keywords=algorithm&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#LinkedInPost](https://www.linkedin.com/feed/hashtag/?keywords=linkedinpost&highlightedUpdateUrns=urn:li:activity:7166316239483461633)  
  
Excited about the progress and challenges ahead!  
   
Make Sure You Follow My GitHub For Solutions: [https://lnkd.in/d7EApJ2m](https://lnkd.in/d7EApJ2m" \t "https://www.linkedin.com/feed/_self)  
   
[https://lnkd.in/dYxzmNTA](https://lnkd.in/dYxzmNTA" \t "https://www.linkedin.com/feed/_self)  
  
Happy coding!

**Solution:-**

class Solution {

    public int findJudge(int n, int[][] trust) {

        int[] inDegree = new int[n + 1];

        int[] outDegree = new int[n + 1];

        // Calculate in-degree and out-degree for each person

        for (int[] relation : trust) {

            outDegree[relation[0]]++;

            inDegree[relation[1]]++;

        }

        // Check for the town judge

        for (int i = 1; i <= n; i++) {

            if (inDegree[i] == n - 1 && outDegree[i] == 0) {

                return i; // Found the town judge

            }

        }

        return -1; // No town judge found

    }

}

