# 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:**

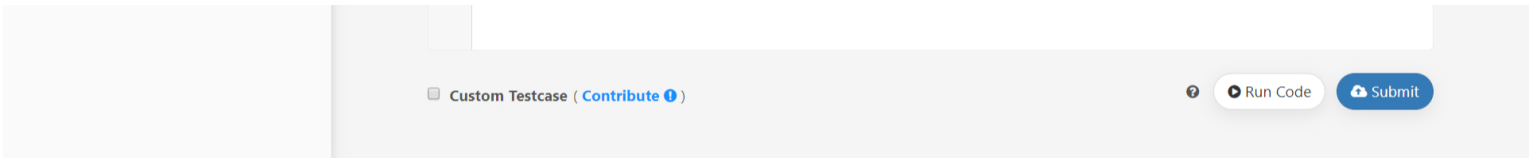
```
Input: N = 4, trust = [[1,3],[1,4],[2,3],[2,4],[4,3]]
Output: 3
```

**Note:**

1. `1 <= N <= 1000`
2. `trust.length <= 10000`
3. `trust[i]` are all different
4. `trust[i][0] != trust[i][1]`
5. `1 <= trust[i][0], trust[i][1] <= N`

Java

```java
class Solution {
    public int findJudge(int N, int[][] trust) {
        if(trust.length==0) return (N==1)?1:-1;
        int[] arr=new int[N+1];
        for(int[] t:trust){
            arr[t[0]]--;
            arr[t[1]]++;
        }
        for(int i=0;i<arr.length;i++){
            if(arr[i]==N-1)
                return i;
        }
        return -1;
    }
}
```


Custom Testcase ( Contribute )
Run Code
Submit