找到小镇的法官

小镇里有 n 个人，按从 1 到 n 的顺序编号。传言称，这些人中有一个暗地里是小镇法官。

如果小镇法官真的存在，那么：

1. 小镇法官不会信任任何人。
2. 每个人（除了小镇法官）都信任这位小镇法官。
3. 只有一个人同时满足属性 **1** 和属性 **2** 。

给你一个数组 trust ，其中 trust[i] = [a_i, b_i] 表示编号为 a_i 的人信任编号为 b_i 的人。

如果小镇法官存在并且可以确定他的身份，请返回该法官的编号；否则，返回 -1 。

示例 1：

输入：n = 2, trust = [[1,2]]

输出：2

示例 2：

输入：n = 3, trust = [[1,3],[2,3]]

输出：3

示例 3：

输入：n = 3, trust = [[1,3],[2,3],[3,1]]

输出：-1

```
class Solution:
    def findJudge(self, n: int, trust: List[List[int]]) -> int:
        indegree = [0 for _ in range(n + 1)]
        outdegree = [0 for _ in range(n + 1)]
        for x, y in trust:
            #对于数组 trust，该数组由信任对 trust[i] = [a, b] 组成，表示编号为 a 的人信任编号为 b 的人。那么 trust[i] = [a, b] 表示从 a 到 b 的有向边，一条从 a 到 b 的有向边，使得点 a 的出度 +1、点 b 的入度 +1。
            outdegree[x] += 1
            indegree[y] += 1

        for x in range(1, n + 1):
            if indegree[x] == n - 1 and outdegree[x] == 0:#法官因为被所有人信
```

```
任，所以其入度为 n-1。不相信任何人，所以其出度为 0
                return x
        return -1
```