Graphs

997. Find the Town Judge

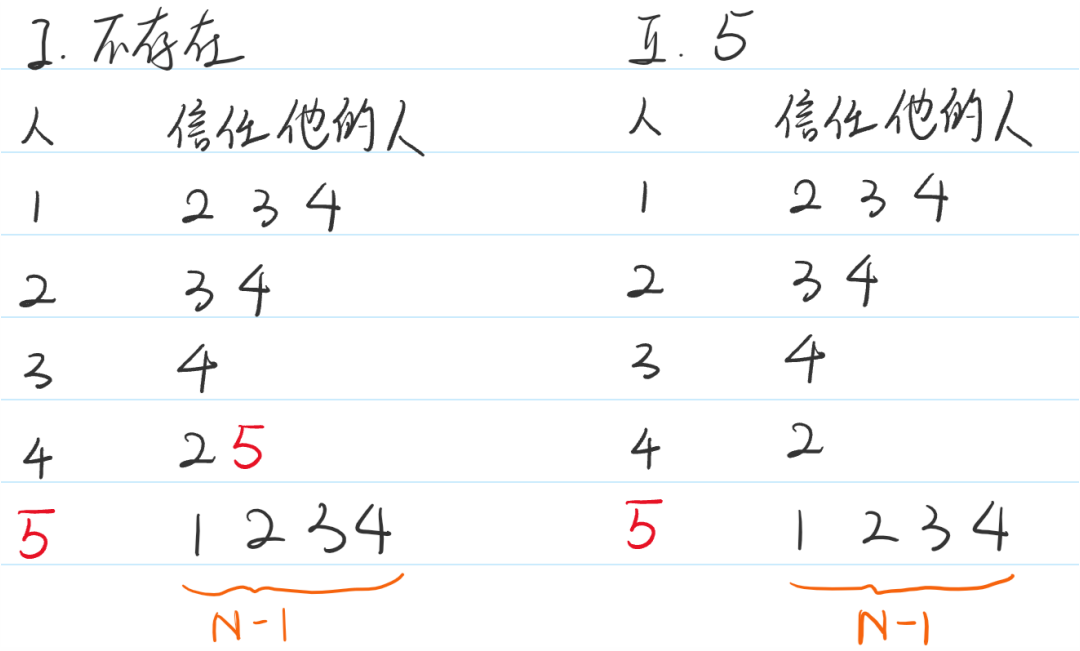
HashMap; Vertex Degree

> 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:
>
> - The town judge trusts nobody.
> - Everybody (except for the town judge) trusts the town judge.
> - 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.

解法一：HashMap

用HashMap保存每个人被他人信任的情况，每个人的编号即为一个Key，值使用ArrayList，一旦遇到有人信任这个人就加入ArrayList并在HashMap中更新。遍历所有编号，如果某个人对应的ArrayList长度是N - 1，则这个人有可能是Town Judge。

但对于此人还需检查他是否出现在别人对应的ArrayList中。



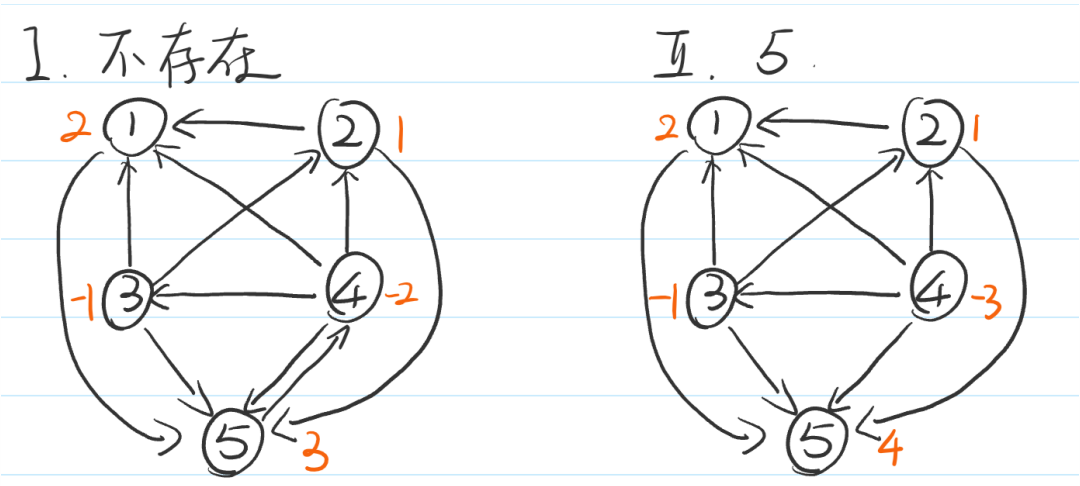
解法二：Graph Vertex Degrees

参考

可以把这道题目想象成一张有向图，对于每个人（结点），如果有人（结点）信任他则将两个结点相连，信任者指向被信任者。最终

每个结点的度数 = 指向其的边 - 其指向外的边

如果有结点度数为N - 1，则是Town Judge




```
class Solution {
    public int findJudge(int N, int[][] trust) {
        int[] ans = new int[N + 1];
        for (int i = 0; i < trust.length; ++i)
        {
            ans[trust[i][1]]++;
```

```
            ans[trust[i][0]]--;
        }
        for (int i = 1; i <= N; ++i)
        {
            if (ans[i] == N - 1)
                return i;
        }
        return -1;
    }
}
```