

277. Find the Celebrity Premium

Medium Topics Companies Hint

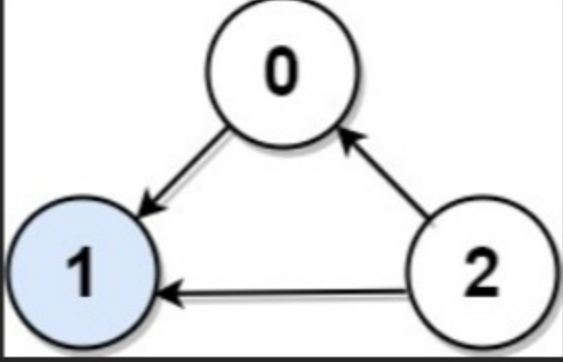
Suppose you are at a party with n people labeled from 0 to $n - 1$ and among them, there may exist one celebrity. The definition of a celebrity is that all the other $n - 1$ people know the celebrity, but the celebrity does not know any of them.

Now you want to find out who the celebrity is or verify that there is not one. You are only allowed to ask questions like: "Hi, A. Do you know B?" to get information about whether A knows B. You need to find out the celebrity (or verify there is not one) by asking as few questions as possible (in the asymptotic sense).

You are given a helper function `bool knows(a, b)` that tells you whether `a` knows `b`. Implement a function `int findCelebrity(n)`. There will be exactly one celebrity if they are at the party.

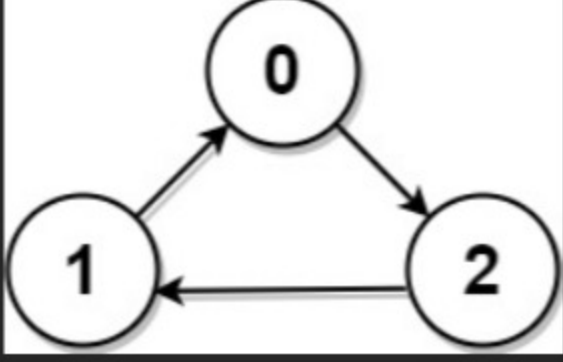
Return the celebrity's label if there is a celebrity at the party. If there is no celebrity, return `-1`.

Example 1:



Input: `graph = [[1,1,0],[0,1,0],[1,1,1]]`
Output: `1`
Explanation: There are three persons labeled with `0`, `1` and `2`. `graph[i][j] = 1` means person `i` knows person `j`, otherwise `graph[i][j] = 0` means person `i` does not know person `j`. The celebrity is the person labeled as `1` because both `0` and `2` know him but `1` does not know anybody.

Example 2:



Input: `graph = [[1,0,1],[1,1,0],[0,1,1]]`
Output: `-1`
Explanation: There is no celebrity.

Constraints:

- `n == graph.length == graph[i].length`
- `2 <= n <= 100`
- `graph[i][j]` is `0` or `1`.
- `graph[i][i] == 1`

Follow up: If the maximum number of allowed calls to the API `knows` is `3 * n`, could you find a solution without exceeding the maximum number of calls?

Seen this question in a real interview before? 1/5

Yes No

Accepted 280.3K | Submissions 588K | Acceptance Rate 47.7%

Topics

Two Pointers Graph Interactive

Companies

0 - 3 months

LinkedIn 6 | TikTok 2 | Toast 2

0 - 6 months

Meta 3 | Microsoft 3

6 months ago

Amazon 3 | DoorDash 3 | Salesforce 2

Hint 1

The best hint for this problem can be provided by the following figure:

```
graph LR; subgraph Scenario1; A1[A] -.-> B1[B]; B1 --> C1[B cannot be the celebrity]; B1 --> D1["A 'can' be the celebrity. Let's move on"]; end; subgraph Scenario2; A2[A] --> B2[B]; B2 --> C2[A cannot be the celebrity]; B2 --> D2["C 'can' be the celebrity. Let's move on"]; end;
```

Hint 2

Well, if you understood the gist of the above idea, you can extend it to find a candidate that can possibly be a celebrity. Why do we say a "candidate"? That is for you to think. This is clearly a greedy approach to find the answer. However, there is some information that would still remain to be verified without which we can't obtain an answer with certainty. To get that stake in the ground, we would need some more calls to the `knows` API.

Similar Questions

Find the Town Judge Easy

Discussion (22)