Problem 277: Find the Celebrity

Problem Information

Difficulty: Medium
Acceptance Rate: 48.72%
Paid Only: Yes
Tags: Two Pointers, Graph, Interactive

Problem Description

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 an integer `n` and 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`.

Note that the `n x n` 2D array `graph` given as input is **not** directly available to you, and instead **only** accessible through the helper function `knows`. `graph[i][j] == 1` represents person `i` knows person `j`, wherease `graph[i][j] == 0` represents person `j` does not know person `i`.

Example 1:

![](https://assets.leetcode.com/uploads/2022/01/19/g1.jpg)

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:

![](https://assets.leetcode.com/uploads/2022/01/19/g2.jpg)

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?

Code Snippets

C++:

```
/* The knows API is defined for you.
bool knows(int a, int b); */

class Solution {
public:
int findCelebrity(int n) {

}
};
```

Java:

```
/* The knows API is defined in the parent class Relation.
boolean knows(int a, int b); */

public class Solution extends Relation {
public int findCelebrity(int n) {
}
}
```

Python3:

```
# The knows API is already defined for you.
# return a bool, whether a knows b
# def knows(a: int, b: int) -> bool:

class Solution:
def findCelebrity(self, n: int) -> int:
```