Description | Solution ▶ | Discuss (456) | Submissions

277. Find the Celebrity

Medium | 👍 2568 | 👎 244 | ♡ Add to List | ⏏ Share

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`.
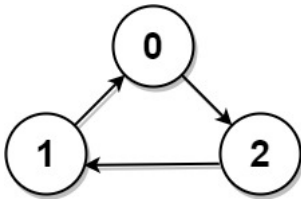
Example 1:
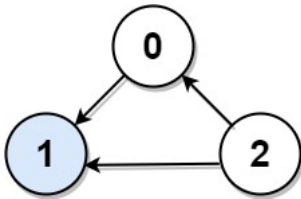

```
Input: graph = [[1,1,0],[0,1,0],[1,1,1]]
Output: 1
Explanation: There are three persons labeled with 0, 1 and 2. graph[i][j] = 1 means
person i knows person j, otherwise graph[i][j] = 0 means person i does not know person j.
The celebrity is the person labeled as 1 because both 0 and 2 know him but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

Accepted 241,392 | Submissions 519,018

Seen this question in a real interview before? | Yes | No

Companies 🔒 ⓘ

Problems | Pick One | ‹ Prev | ⚙/150 | Next ›

C++ ▾ | Autocomplete | ⓘ | {} | ⟳

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

class Solution {
public:
    int findCelebrity(int n)
    {
        int low = 0;
        int high = n-1;

        while(low<high)
        {
            if(knows(low,high))
            {
                low++;
            }
            else
            {
                high--;
            }
        }
        cout<<low<<" "<<high<<" "<<n;
        for(int i=0;i<n;i++)
        {
            if(i!=low)
                if(knows(low,i) || !knows(i,low))
                    return -1;
        }
        return low;
    }
};
```

NEW

Testcase | Run Code Result | Debugger 🔒 | ∨

Accepted Runtime: 4 ms ⓘ

Your input | `[[1,0],[0,1]]`

stdout | `0 0 2`

Output | `-1` | | Diff

Expected | `-1`

xample ⓘ ases | 🔒 Run Code ^ | Submit