● Autocomplete

277. Find the Celebrity
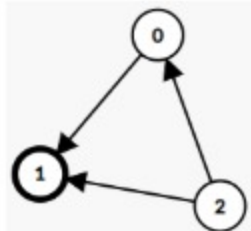
Medium   👍 993   👎 127   ♡ Add to List   🔗 Share

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 him/her but he/she does not know any of them.

Now you want to find out who the celebrity is or verify that there is not one. The only thing you are allowed to do is to ask questions like: "Hi, A. Do you know B?" to get information of 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)` which tells you whether A knows B. Implement a function `int findCelebrity(n)`. There will be exactly one celebrity if he/she is in the party. Return the celebrity's label if there is a celebrity in the party. If there is no celebrity, return `-1`.
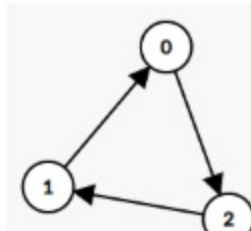
Example 1:


```
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.
```

Note:

1. The directed graph is represented as an adjacency matrix, which is an `n x n` matrix where `a[i][j] = 1` means person `i` knows person `j` while `a[i][j] = 0` means the contrary.
2. Remember that you won't have direct access to the adjacency matrix.

Accepted 116,720 | Submissions 285,921

Seen this question in a real interview before?   Yes   No

Contributor

Companies 🔒 ⓘ

Related Topics

Similar Questions

Show Hint 1

Show Hint 2

```python
# The knows API is already defined
for you.
# @param a, person a
# @param b, person b
# @return a boolean, whether a
knows b
# def knows(a, b):

class Solution(object):
    def findCelebrity(self, n):
        """
        :type n: int
        :rtype: int
        """
```