LeetCode 277

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 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), your function should minimize the number of calls to knows.

Note: 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.

Thought

Because celeb knows no one and every one knows celeb, so we can use a for loop to check if the adjacent people(a,b) knows each other, and at the same time update celeb to b if a knows b. But that only verify b does not know ppl in [b+1, n-1], so we need another for loop to check b with ppl in [0, b-1]

Solution

```
public class Solution extends Relation {
  public int findCelebrity(int n) {
      int celeb = 0;
      for(int i = 1; i < n; i++){
          if(knows(celeb, i))
              celeb = i;
      }
      for(int j = 0; j < celeb; j++){
          if(j != celeb && (knows(celeb, j) || !knows(j, celeb))) return -1;
      }
      return celeb;
  }
}
```

Takeaways

- Visualize the problem