problem there are n people at a party. any one person may or may not know any other person. (but A can know B without B knowing A). a celebrity is a person that knows no one, but everyone knows of. you arrive at the party with the intent of finding all celebrities. you may ask any person whether they know any other person. how can you achieve your goal efficiently? (asking a minimal number of questions).

solution 1 first, we note that there can be at most one celebrity. the main observation is that whenever we ask a person A whether they know person B, there are two cases: A does not know B, which implies B is not a celebrity, or A knows B, which implies A is not a celebrity. in any case, the answer lets us reduce the pool of possible celebrity candidates by one. starting with the whole n people as celebrity candidates, we may, after $n-1$ questions, arrive at a unique candidate for a celebrity, which we call C. at worst, it takes another $2(n-1)-1$ questions to find out if C is indeed a celebrity, for a total of $3n-4$ questions. (where we of course assumed $n \geq 2$)

solution 2 again, we note that there can be at most one celebrity at the party. also, if C is a celebrity at the party, then C is also a celebrity at any subparty. so we may recursively solve this problem as follows.

```
def find_celebrities(party):
    if |party| = 1:
        return party
    partition party = sub_party1 ⊔ sub_party2 where |sub_party_1| = |party| / 2
    let celebrity_candidates = find_celebrities(sub_party1) ∪ find_celebrities(sub_party2)
    for cc in celebrity_candidates:
        if is_celebrity(cc, party):
            return {cc}
    return {}
```

this solution asks $O(n \log n)$ questions at worst.