Asking the Important Questions

Since coming to UC Irvine, has anyone met a celebrity?
What is a celebrity?

- Within a group of people $G$, we say a person $p$ is a celebrity (famous) if:
  - Everyone knows who $p$ is (celebrities must be known by everyone)
  - Person $p$ does not know who anyone else is

- Suppose I think person $p$ might be a celebrity within group $G$.
  - How can you check?
  - What actions can we take?

*Ask: does $a$ know $b$? (for $a,b \in G$ select)*
Who is a celebrity?

- Given group (of people) \( G \) and person \( p \)
  How efficiently can I check if person \( p \) is a celebrity?

```python
for each \( x \in G - \{p\} \)
  if \( x \) doesn't know \( p \)
    return false
  if \( p \) knows \( x \)
    return false
return true
```

- How can I check if a group \( G \) has any celebrity?

  for each \( p \in G \), check as above
Finding a Celebrity

- The previous solution was “brute force.”
- Can we do better?
- Things to look for:
  - Did we repeat work?
    (work that may have been needed, but was done twice)
  - Did we do unnecessary work?
    (work that we could have done without)
Finding a Celebrity

Given group $G$, is there a celebrity, and if so, who?

$G' =$ copy of $G$ // "standing" in metaphor

while $|G'| > 1$

$a_1 b =$ two arbitrary distinct elements of $G'$

if $a$ knows $b$

remove $a$ from $G'$

$n - 1$

else

remove $b$ from $G'$

$x =$ remaining one in $G'$, check if $x$ celebrity

$2(n-1)$
Finding a Celebrity

- Previous solution assumes a group has at most one celebrity.

- Could a group have two or more celebrities?

Suppose FSOC yes

G has two celebs. Call them a,b

Does a know b?

If yes \[ \rightarrow \langle \]

If no \[ \rightarrow \langle \]

What’s Next?

● Wednesday: dynamic programming
  ○ Probably something you saw as an undergraduate
  ○ If not, please read the introduction section in textbook
    ■ Or “Fibonacci Sequence” section of “Examples: Computer Algorithms” at Wikipedia
  ○ I do not assume you took algorithms last quarter.

● In general, this class extends and reinforces your ugrad algorithms
  ○ How did I come up with this class?