Given a problem:
Is there a solution?
What is it?
How good is it?

Write as many definitions of “good” (for solutions to CS problems) as you can.
How many prints?

This is not a trick question

print 42
How many prints?

```python
for i in range(n):
    print 42
```
How many prints?

```python
for i in range(n):
    for j in range(n):
        print 42
```

```python
for i in range(n):
    for j in range(m):
        print 42
```

```python
for i in range(n):
    for j in range(i):
        print 42
```
How many calls?

def fact(n):
    if n == 0:
        return 1
    return n * fact(n-1)
How many calls?

def fib(n):

    if n == 1 or n == 2:
        return 1

    return fib(n−2) + fib(n−1)
“Big-O”: the colloquial definition

an upper bound on the performance of an algorithm, when the inputs to that algorithm are very large
“Big-O”: the formal definition

“Big O” is notation from the mathematical subfield of asymptotic analysis.

Asymptotic analysis gives us a way to compare the behavior of functions at their limits.

Informally, we say that a function $f$ is “in” $O(g)$ if $f(n)$ is always bounded above by $g(n)$, once $n$ is sufficiently large.

More formally:

$$f(n) \in O(g(n)) \text{ if and only if there exist positive constants } c \text{ and } N \text{ such that } 0 \leq (f(n)) \leq c \cdot g(n) \text{ for all } n \geq N$$
Sorting!