Daily Coding Problem #233

Problem

This problem was asked by Apple.

Implement the function `fib(n)`, which returns the nth number in the Fibonacci sequence, using only `O(1)` space.

Solution

Finding the nth Fibonacci sequence is a classic problem with a variety of solutions.

Recall that the Fibonacci sequence can be defined such that the first two numbers are `0` and `1`, and thereafter each number is the sum of the two that came before it. The first few numbers are `0, 1, 1, 2, 3, 5, ....`

Because of the property that each number depends only on the two preceding numbers, it is straightforward to use recursion to find the nth Fibonacci number.

```
def fib(n: int):
    if n <= 1:
        return n
    else:
        return fib(n - 1) + fib(n - 2)
```

As n gets very large, however, this quickly becomes inefficient. The reason for this is that every call to `fib` generates two additional calls, meaning that we must invoke this function on the order of 2^n times.

We can reduce the number of calls by saving previous results in a cache. If we have already calculated `fib(n)` for some arbitrary n, we return the value stored in our cache. If not, we make sure to set the value in the cache before returning.

```
cache = {0: 0, 1: 1}

def fib(n: int):
    if n in cache:
        return cache[n]
    else:
        cache[n] = fib(n - 1) + fib(n - 2)
        return cache[n]
```

Neither of these solutions satisfy the requirement that we only use constant space. Note that each number in the sequence only depends on the two preceding ones. So we can find the n^{th} Fibonacci number by continually updating these two numbers, according to the formula `a, b = b, a + b`. For example, 3 and 5 will turn into 5 and 8. Once the updates are done, we simply take the last number.

```
def fib(n: int):
    if n <= 1:
        return n
    else:
        a, b = 0, 1
        for _ in range(n - 1):
            a, b = b, a + b
        return b
```

Finally, it is in fact possible to write an `O(1)` time *and* space algorithm, since there is a closed form mathematical solution. While you probably will not be expected to provide this in an interview, it may still be interesting to know.

```
from math import sqrt

PHI = (1 + sqrt(5)) / 2

def fib(n: int):
    return int(PHI ** n / sqrt(5) + 0.5)
```