Example: Fibonacci

```
def fibRecursive(n):
    if n <= 1:
        return 1
    return fibRecursive(n-1) + fibRecursive(n-2)

def fibIterative(n):
    a, b = 1, 1
    for _ in range(n):
        a, b = b, a+b
    return a
```

The recursive version runs in $O(2^n)$ time, but the iterative version is $O(n)$.

Space complexity

- So far, have only talked about time complexity
- Space complexity: amount of space that program uses
- Example with O(n^2) space complexity (and O(n^2) runtime):

```
A = [ ]
for i in range(n):
    for j in range(n):
        A.append(1)
```