

CS21: INTRODUCTION TO COMPUTER SCIENCE

Prof. Mathieson

Fall 2018

Swarthmore College

Outline Dec 5:

- Recursion (example: Fibonacci)
- Stack diagrams for recursion
- Start: graphics examples
- Go over Quiz 5

Notes

- Lab 11 is due Sunday night
- Ninja session tonight and Friday
- Office hours on Friday

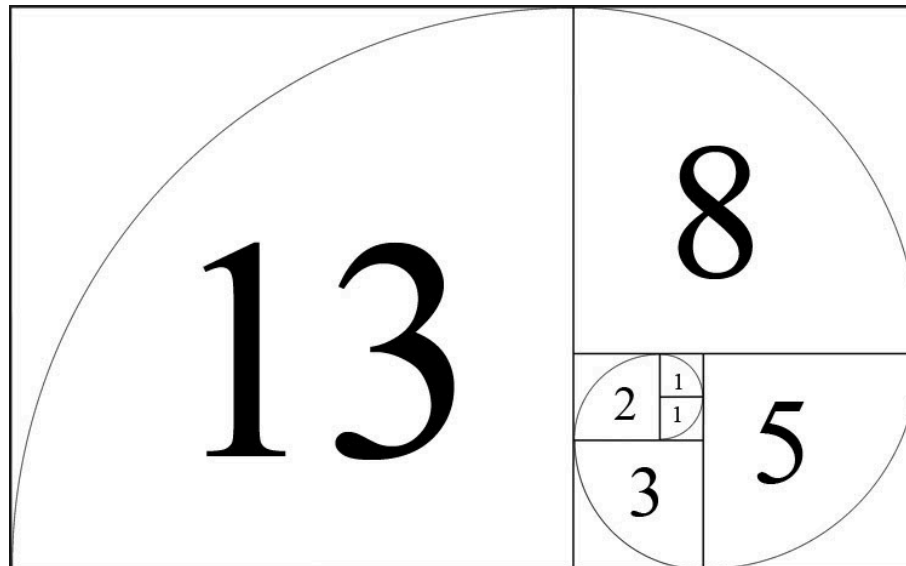
Fibonacci Example

Fibonacci numbers

Each Fibonacci number is the sum of the previous two Fibonacci numbers

Recursion: $F_n = F_{n-1} + F_{n-2}$

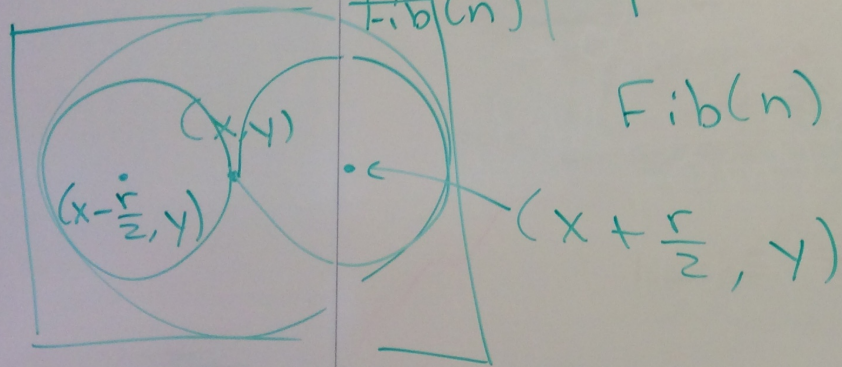
Base cases: $F_0 = 1$ and $F_1 = 1$



Fibonacci Numbers

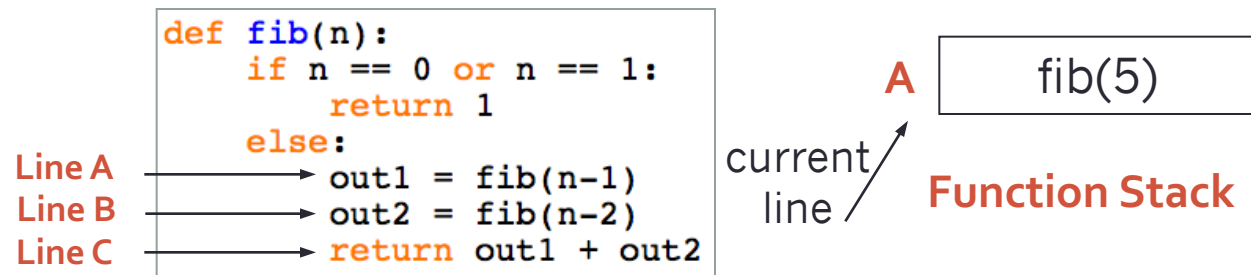
n	0	1	2	3	4	5	6	7	...
Fib(n)	1	1	2	3	5	8	13	21	...

$$Fib(n) = Fib(n-1) + Fib(n-2)$$

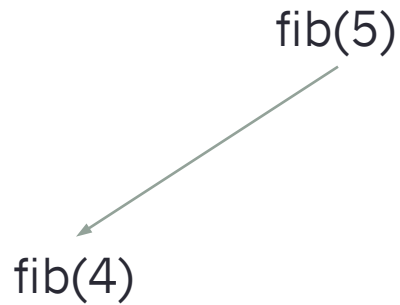


Fibonacci Function Stack

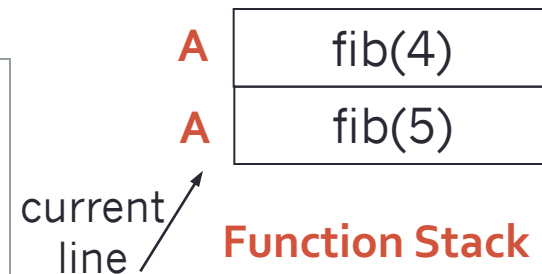
fib(5)



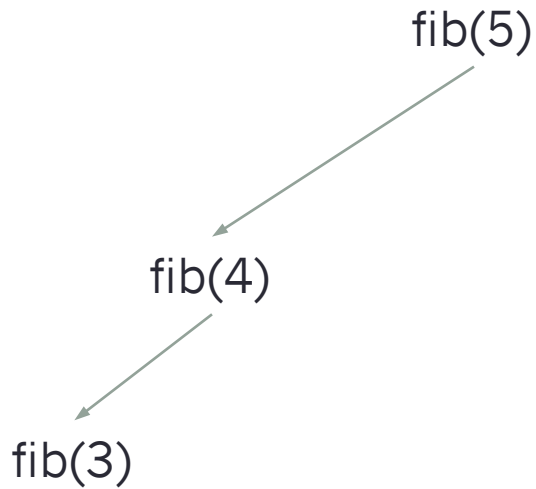
Fibonacci Function Stack



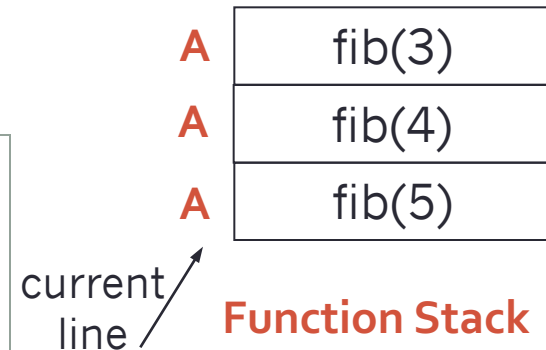
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



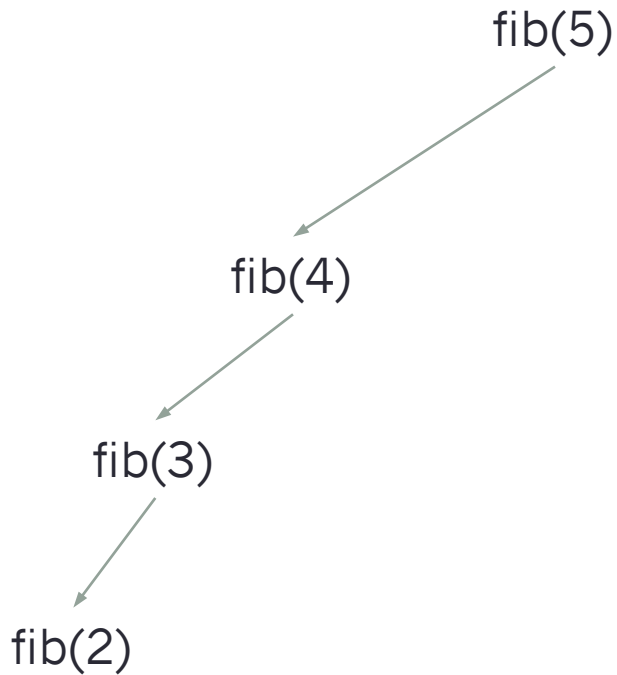
Fibonacci Function Stack



```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



Fibonacci Function Stack



```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        out1 = fib(n-1)  
        out2 = fib(n-2)  
        return out1 + out2
```

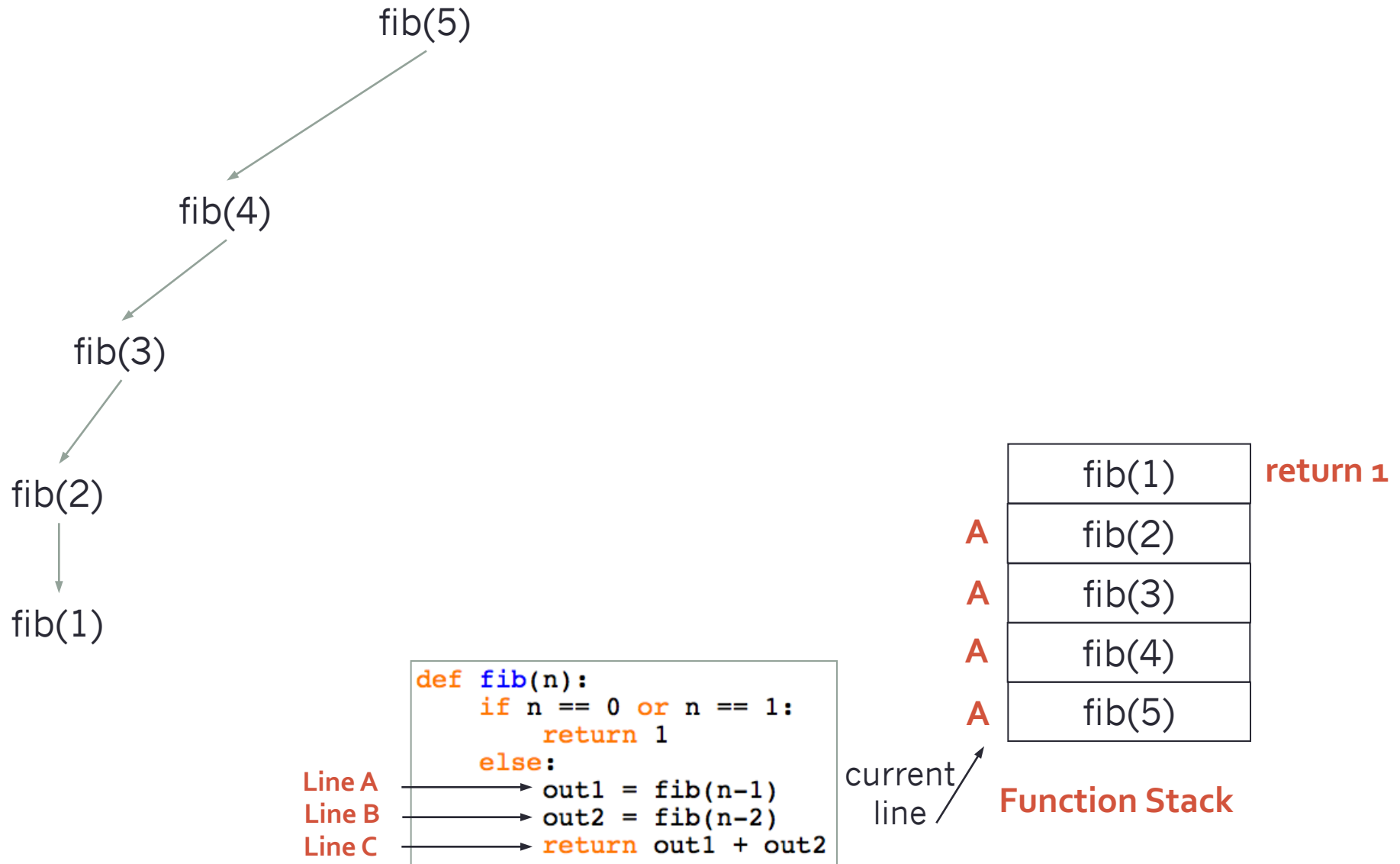
Line A →
Line B →
Line C →

A	fib(2)
A	fib(3)
A	fib(4)
A	fib(5)

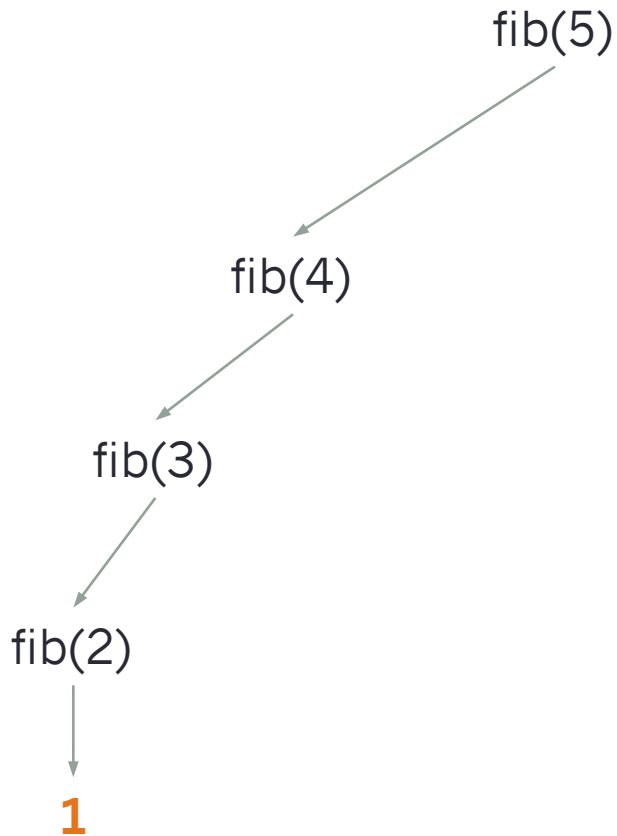
current
line ↗

Function Stack

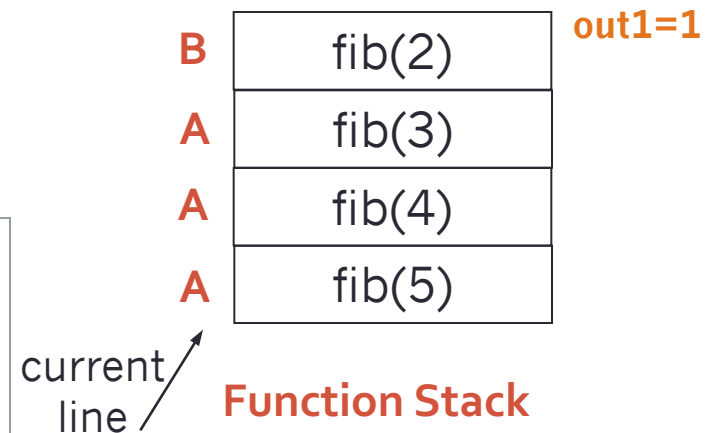
Fibonacci Function Stack



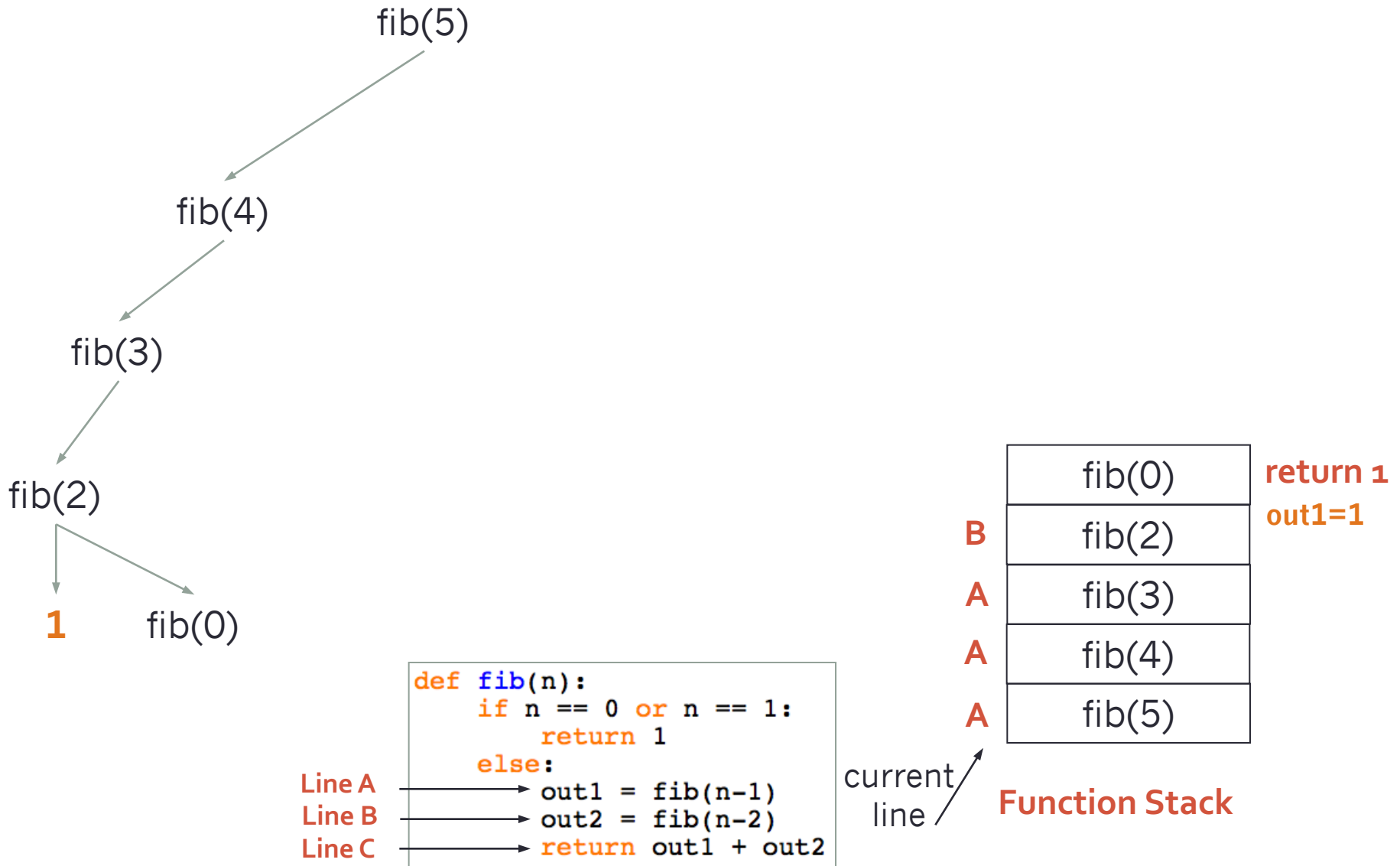
Fibonacci Function Stack



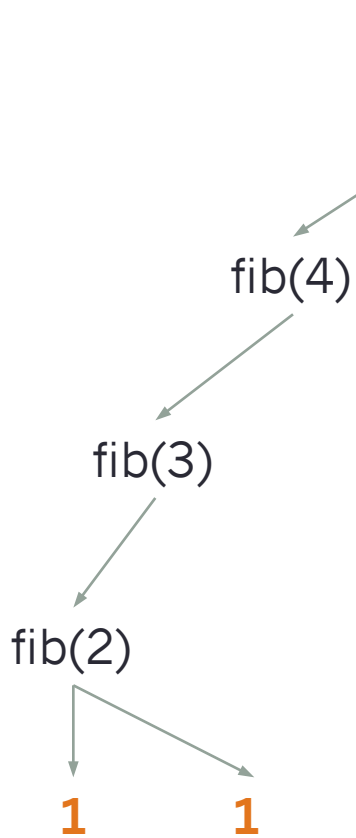
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



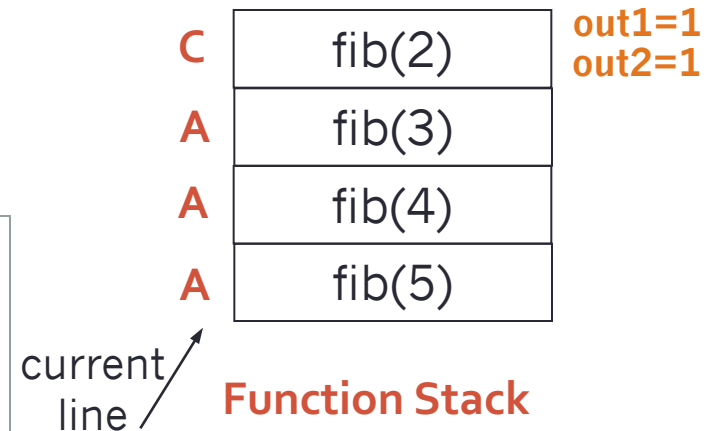
Fibonacci Function Stack



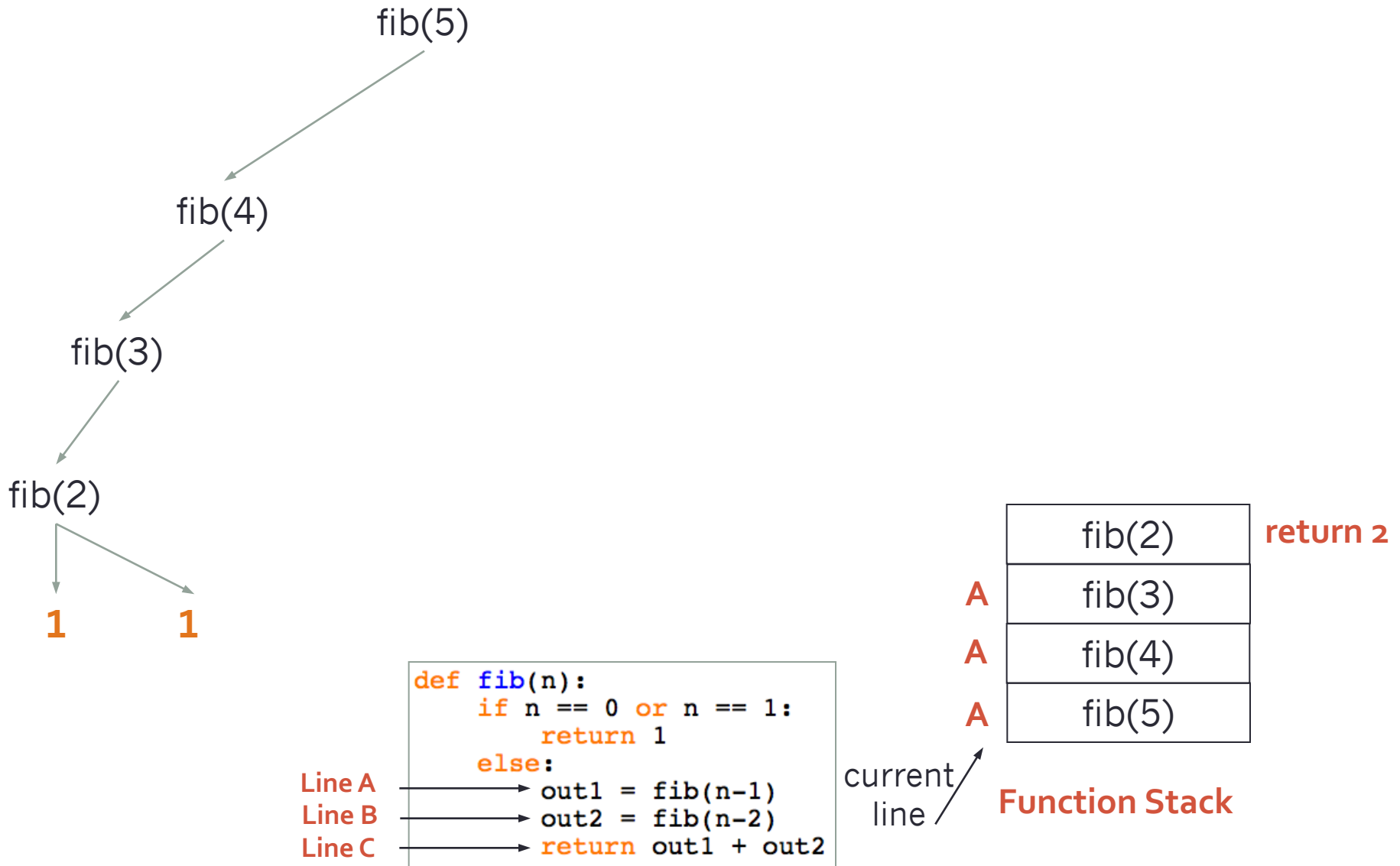
Fibonacci Function Stack



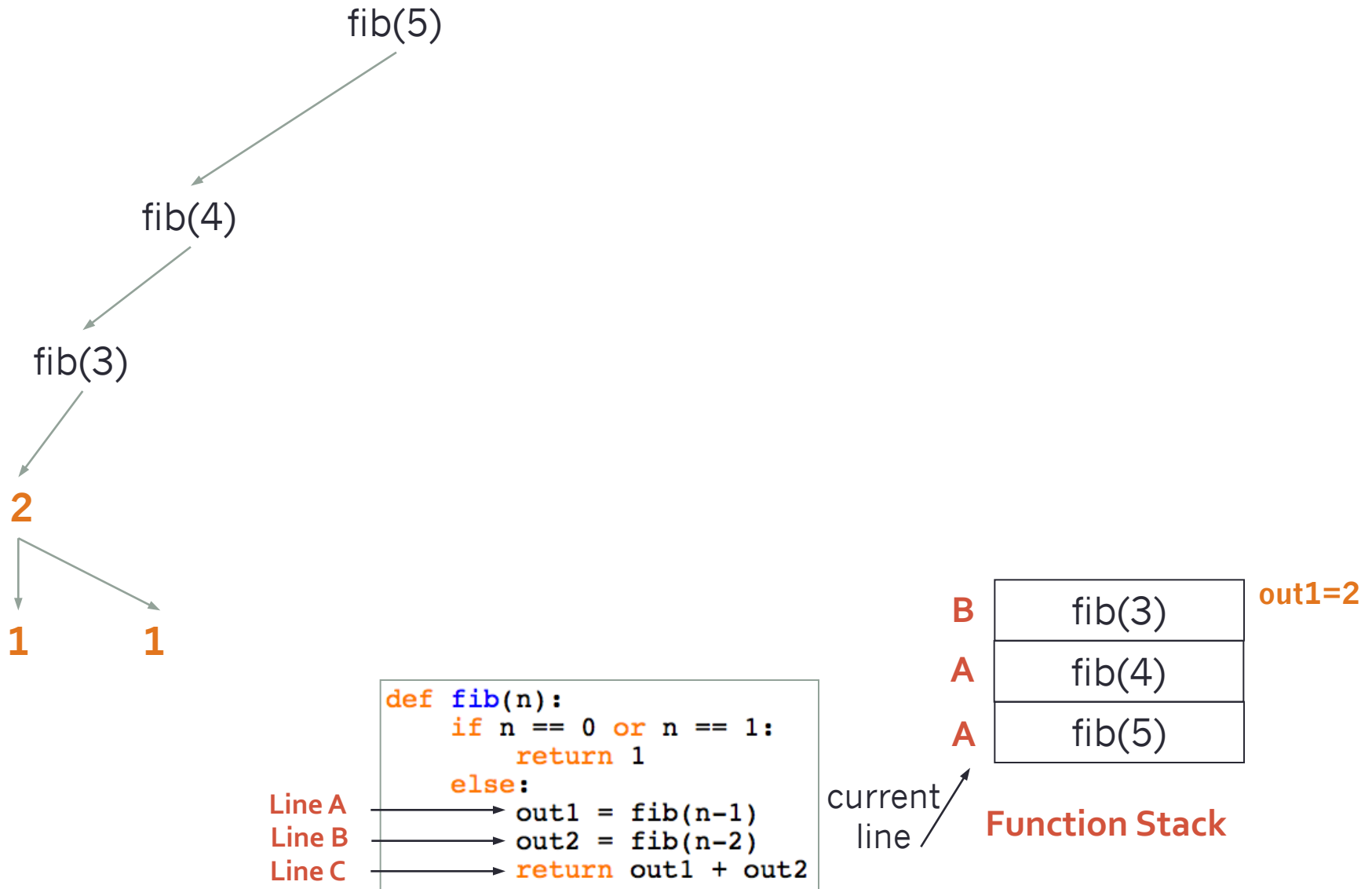
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



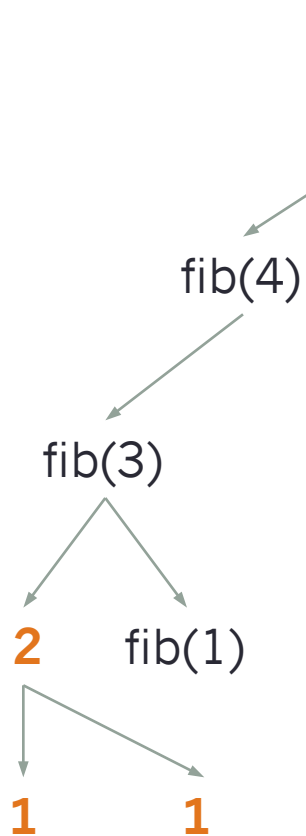
Fibonacci Function Stack



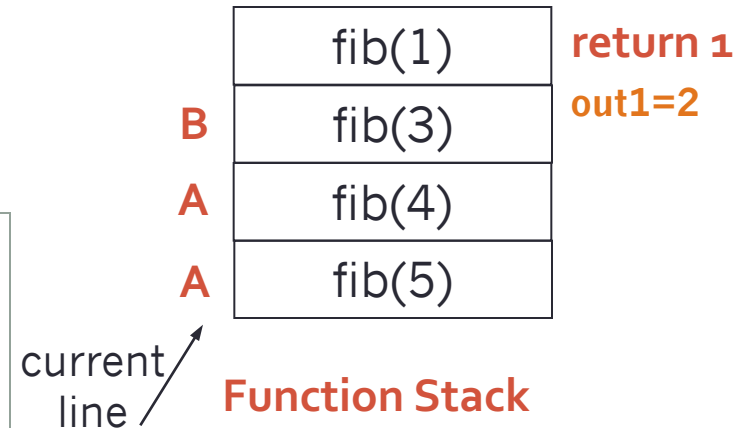
Fibonacci Function Stack



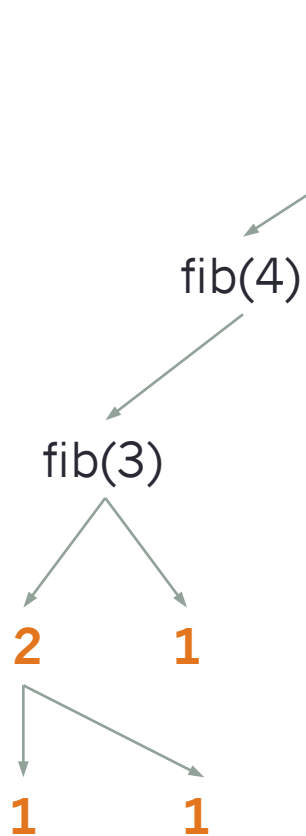
Fibonacci Function Stack



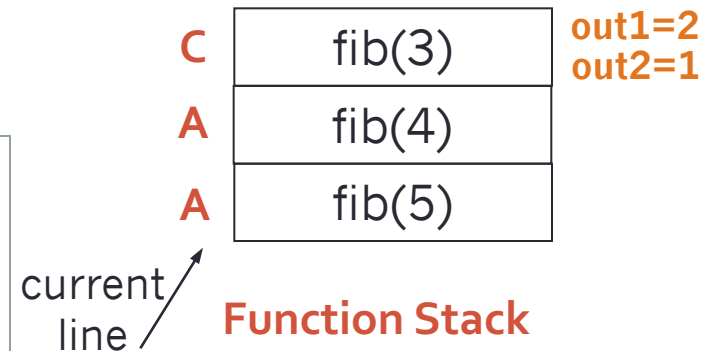
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



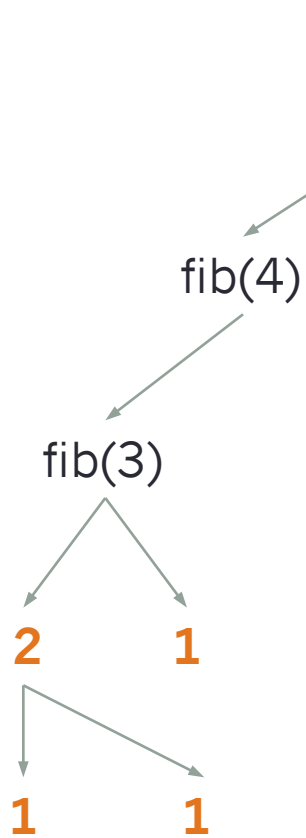
Fibonacci Function Stack



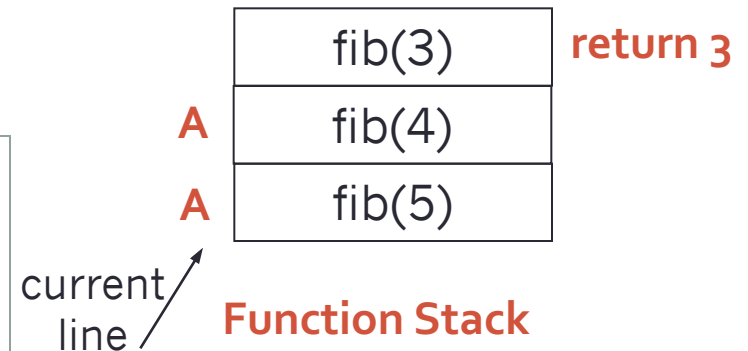
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



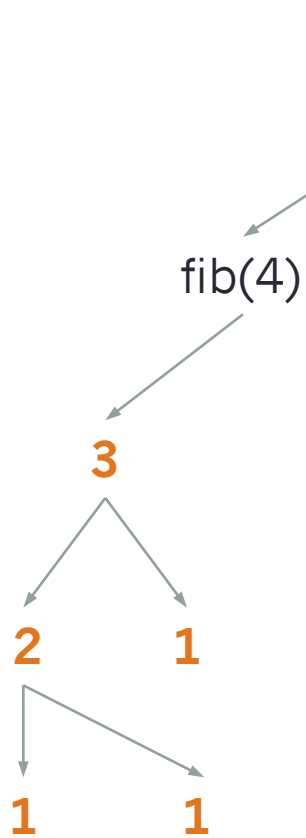
Fibonacci Function Stack



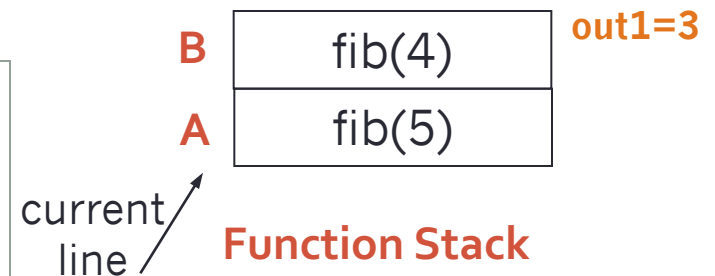
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



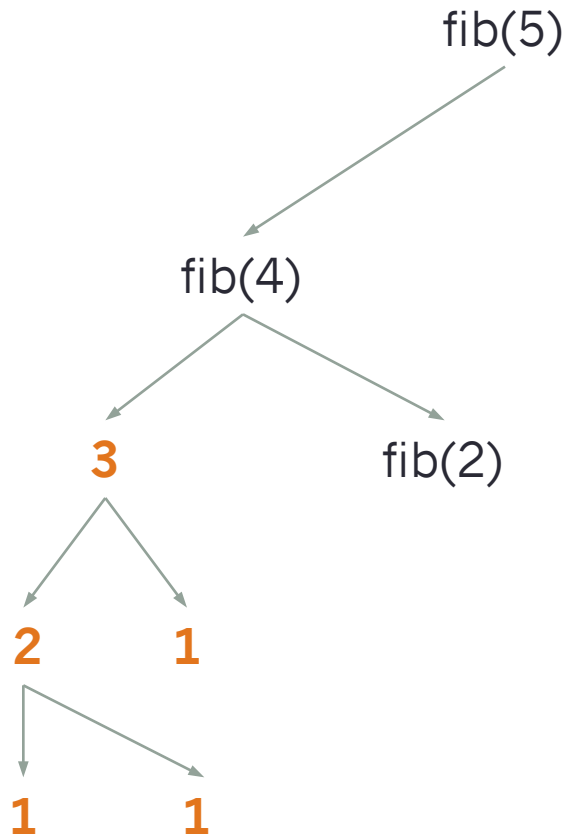
Fibonacci Function Stack



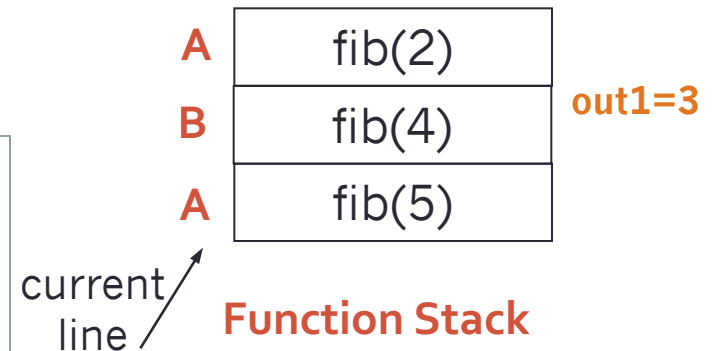
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



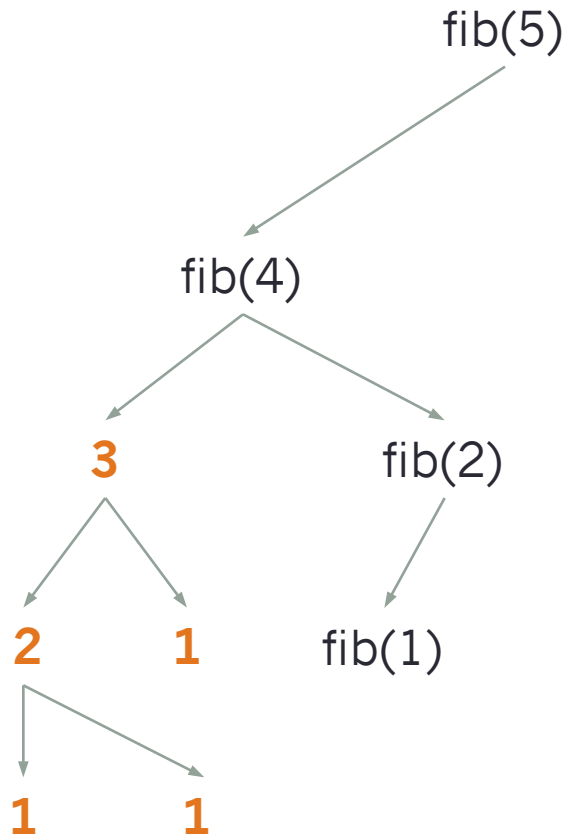
Fibonacci Function Stack



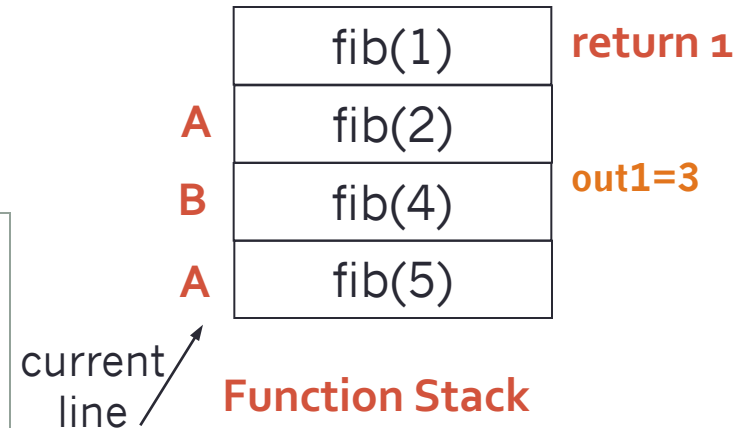
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



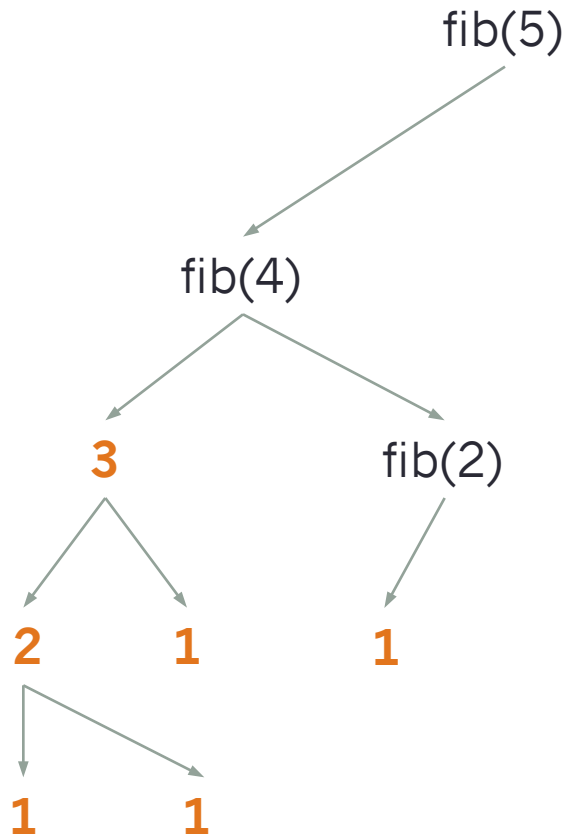
Fibonacci Function Stack



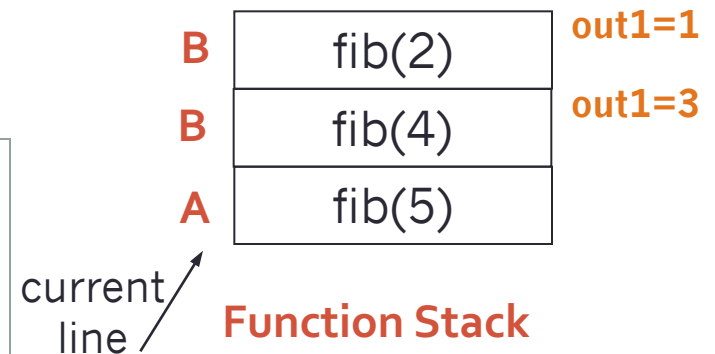
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



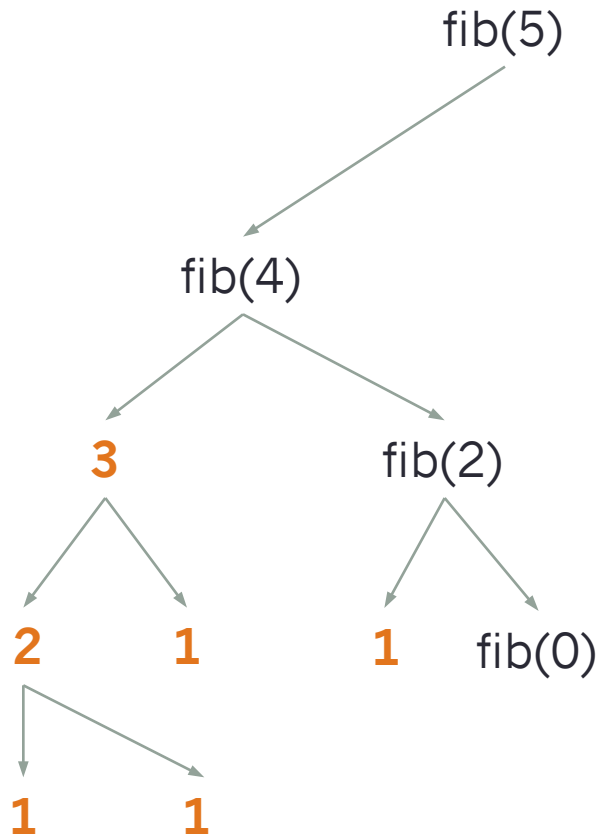
Fibonacci Function Stack



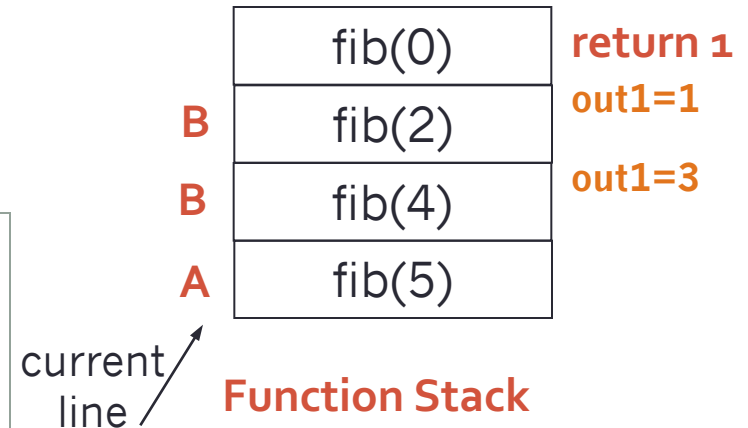
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



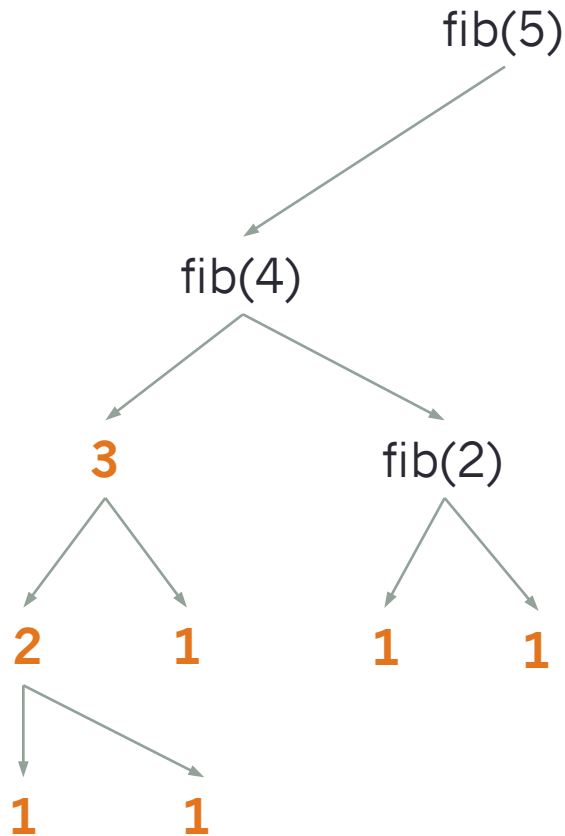
Fibonacci Function Stack



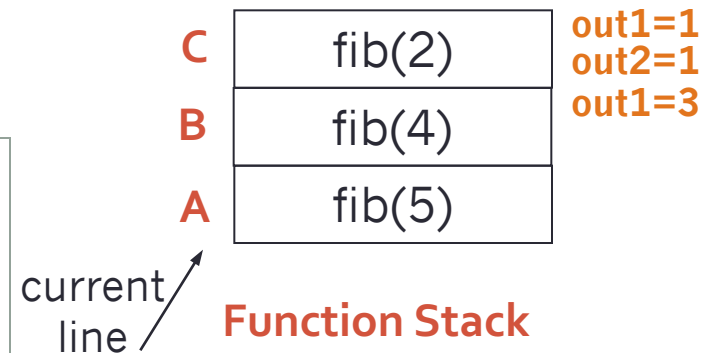
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



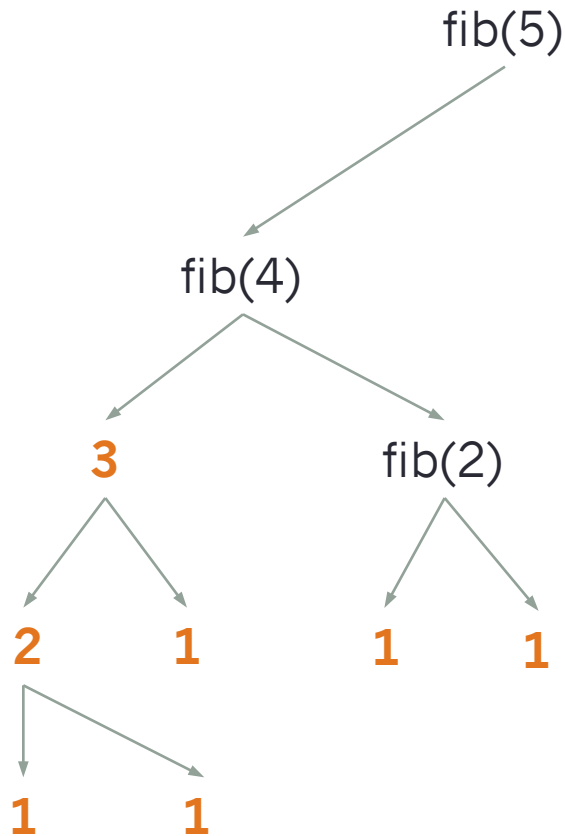
Fibonacci Function Stack



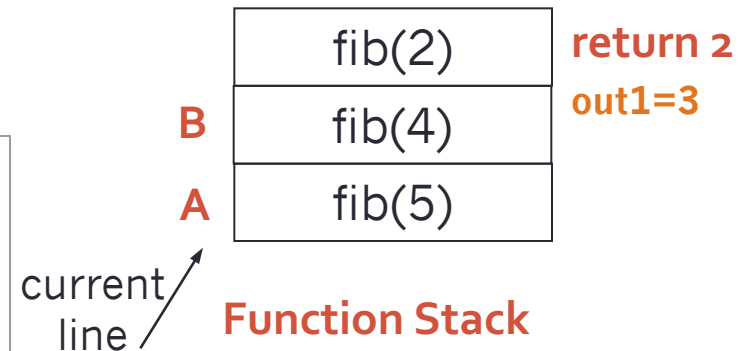
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



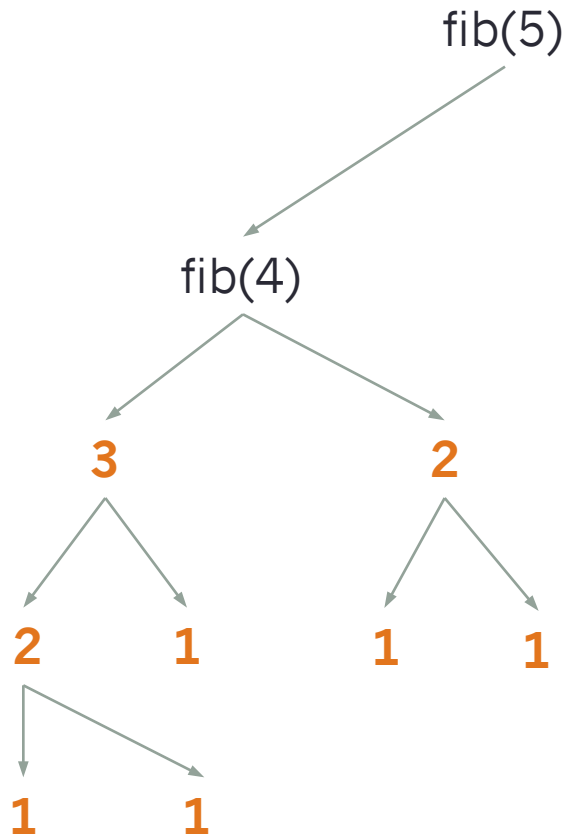
Fibonacci Function Stack



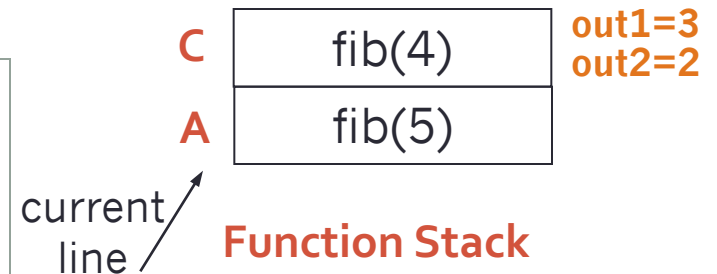
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



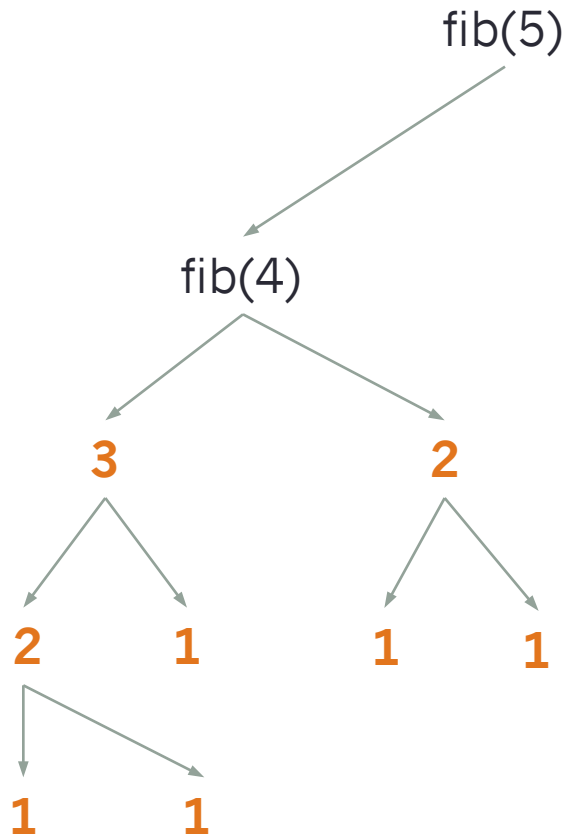
Fibonacci Function Stack



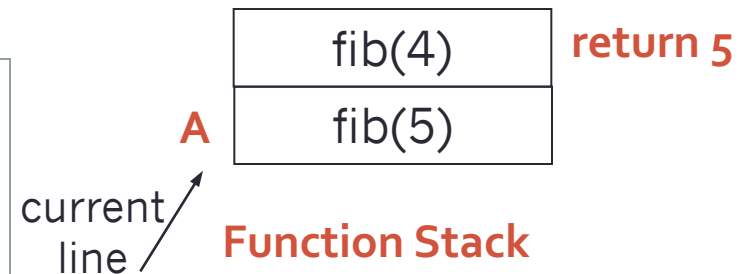
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



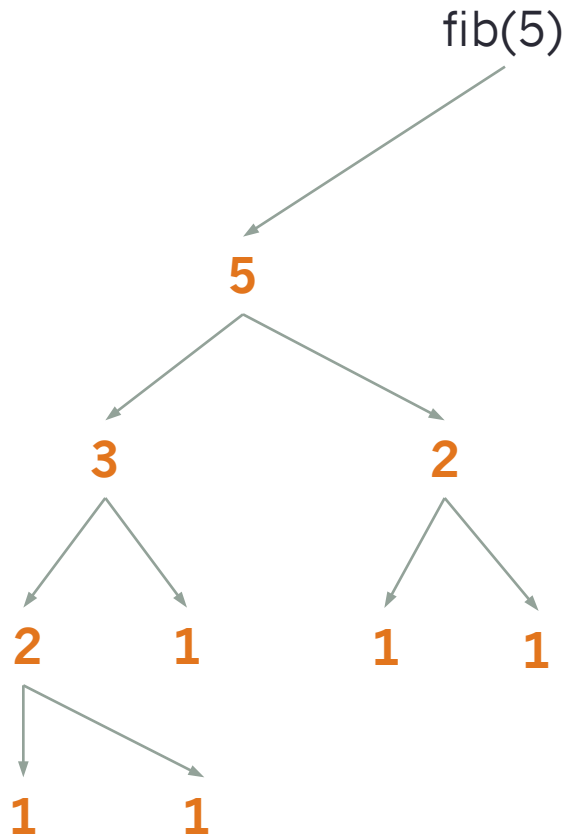
Fibonacci Function Stack



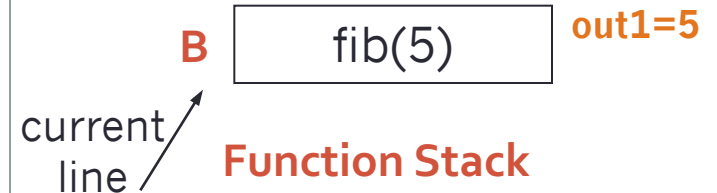
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



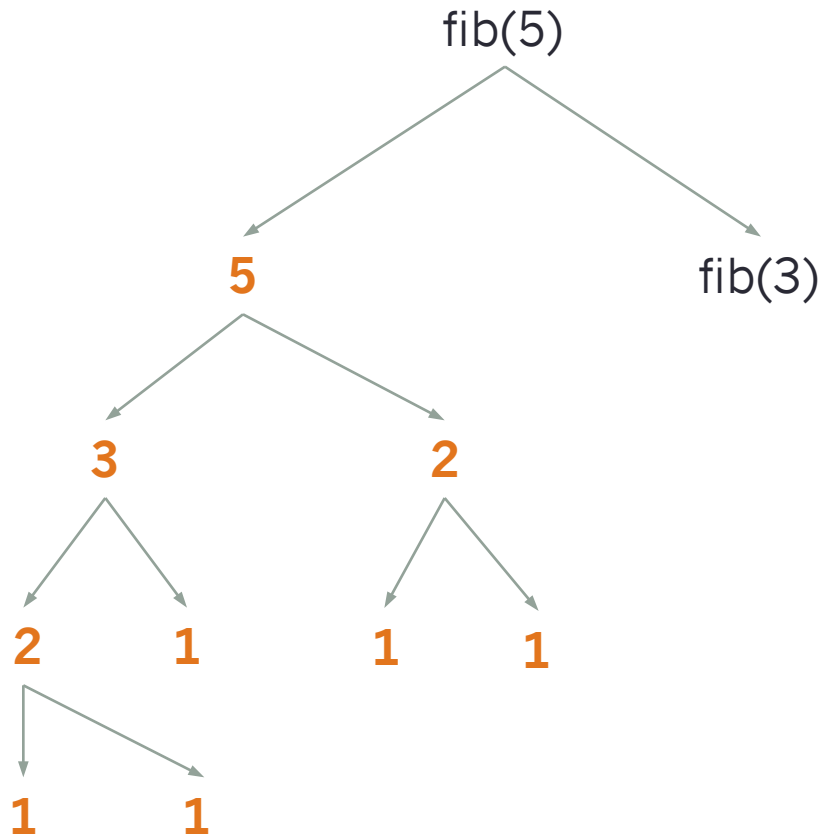
Fibonacci Function Stack



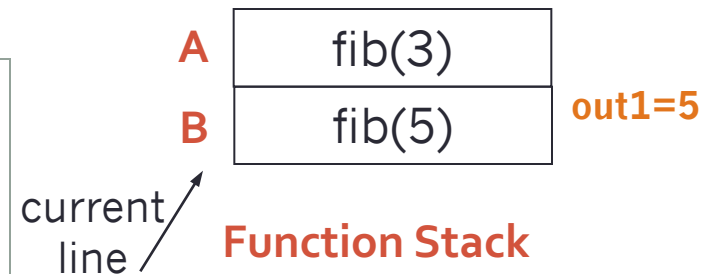
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



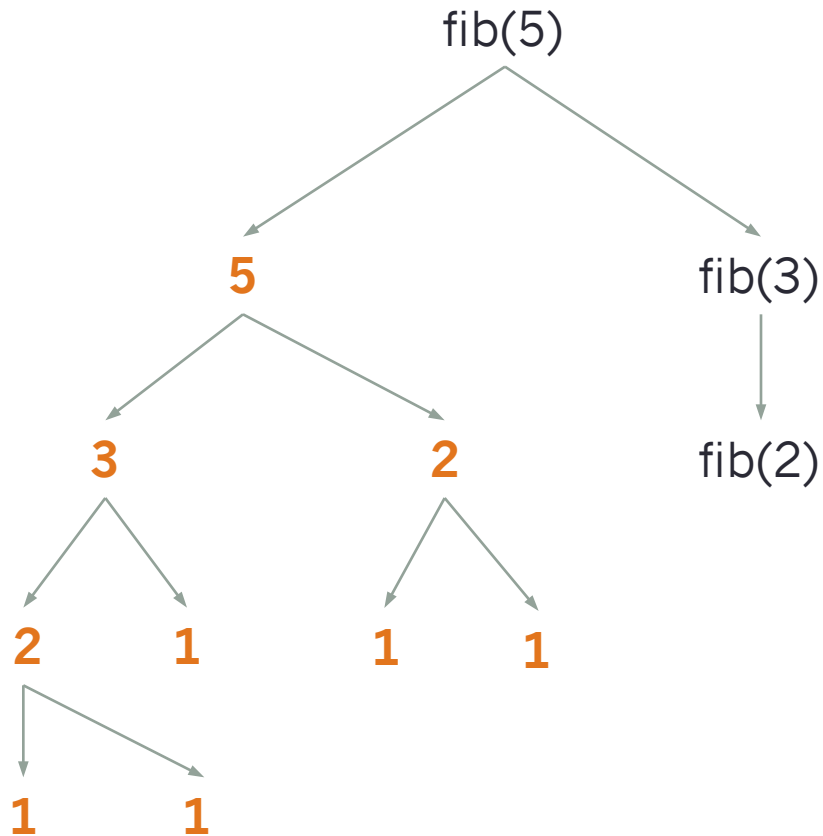
Fibonacci Function Stack



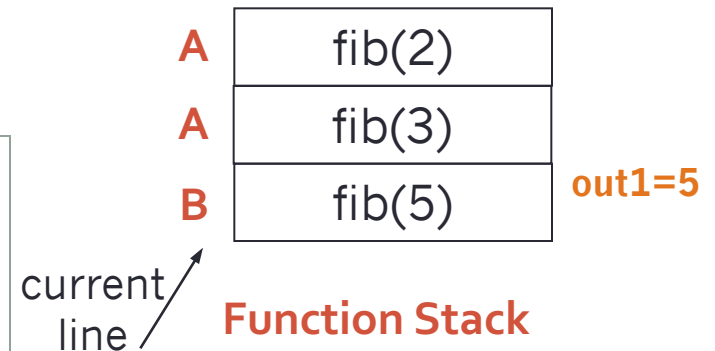
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



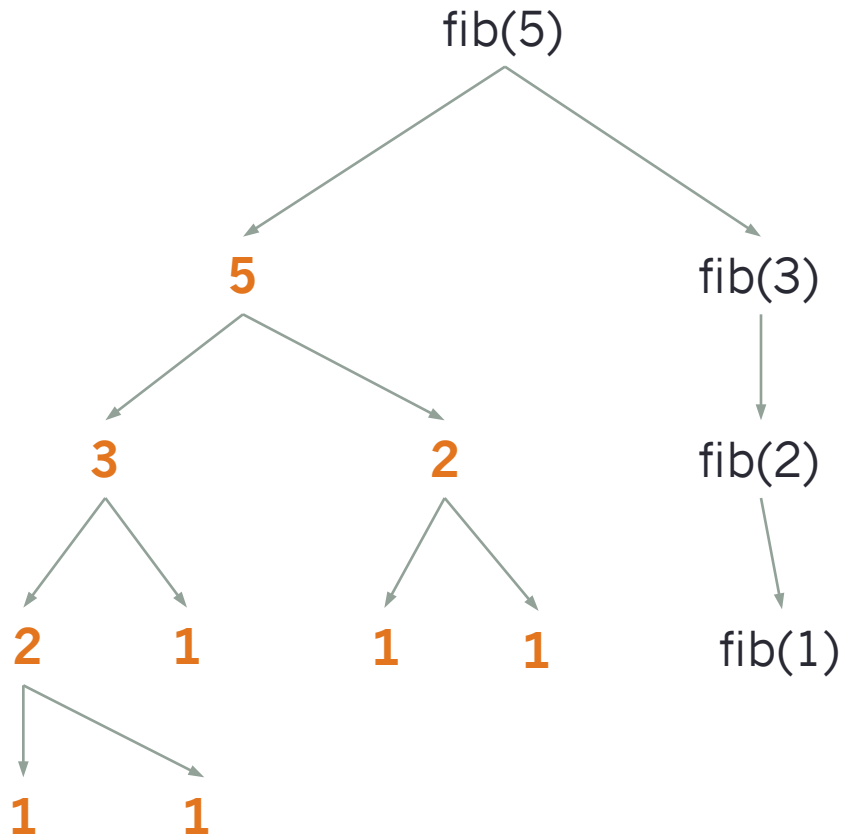
Fibonacci Function Stack



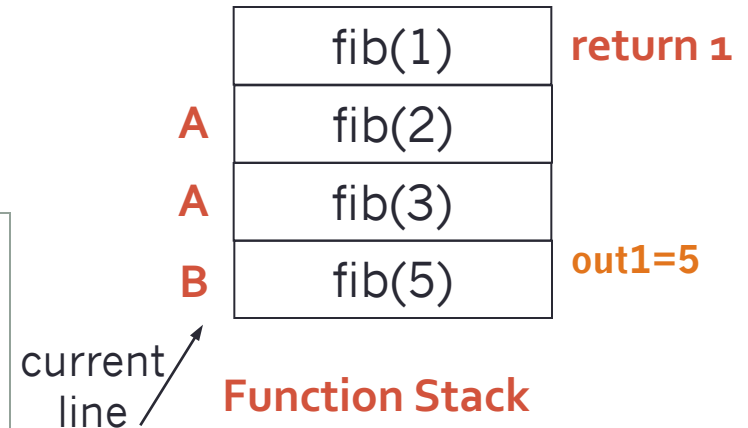
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



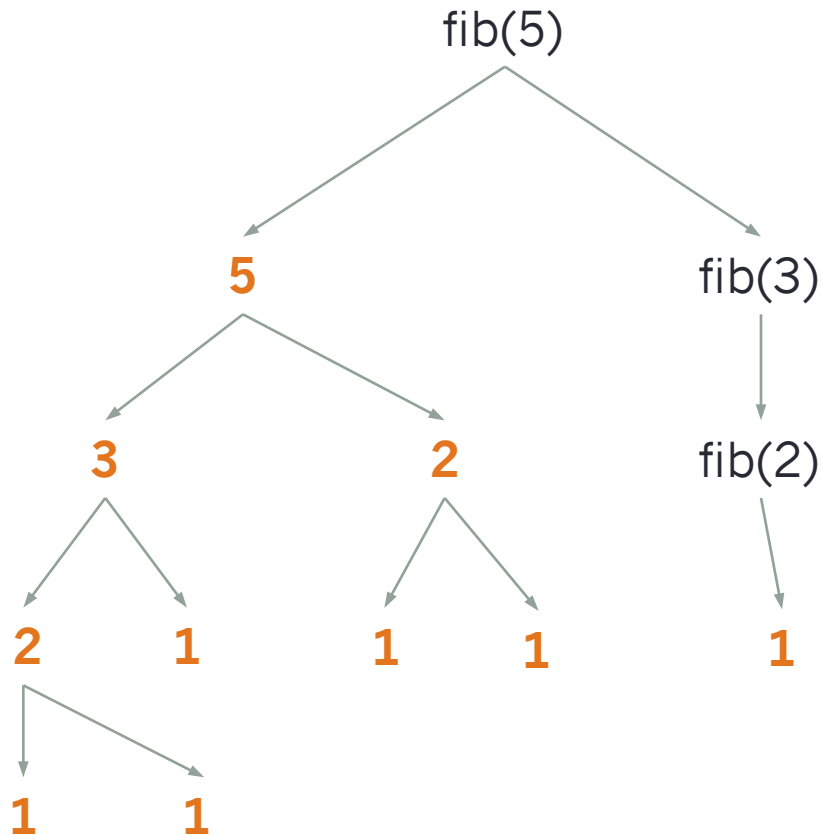
Fibonacci Function Stack



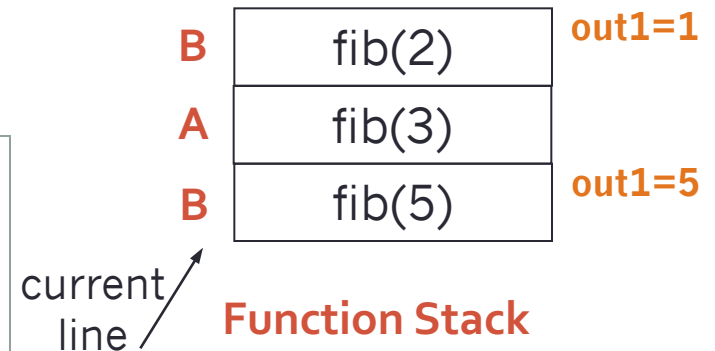
```
def fib(n):
    if n == 0 or n == 1:
        return 1
    else:
        out1 = fib(n-1)
        out2 = fib(n-2)
        return out1 + out2
```



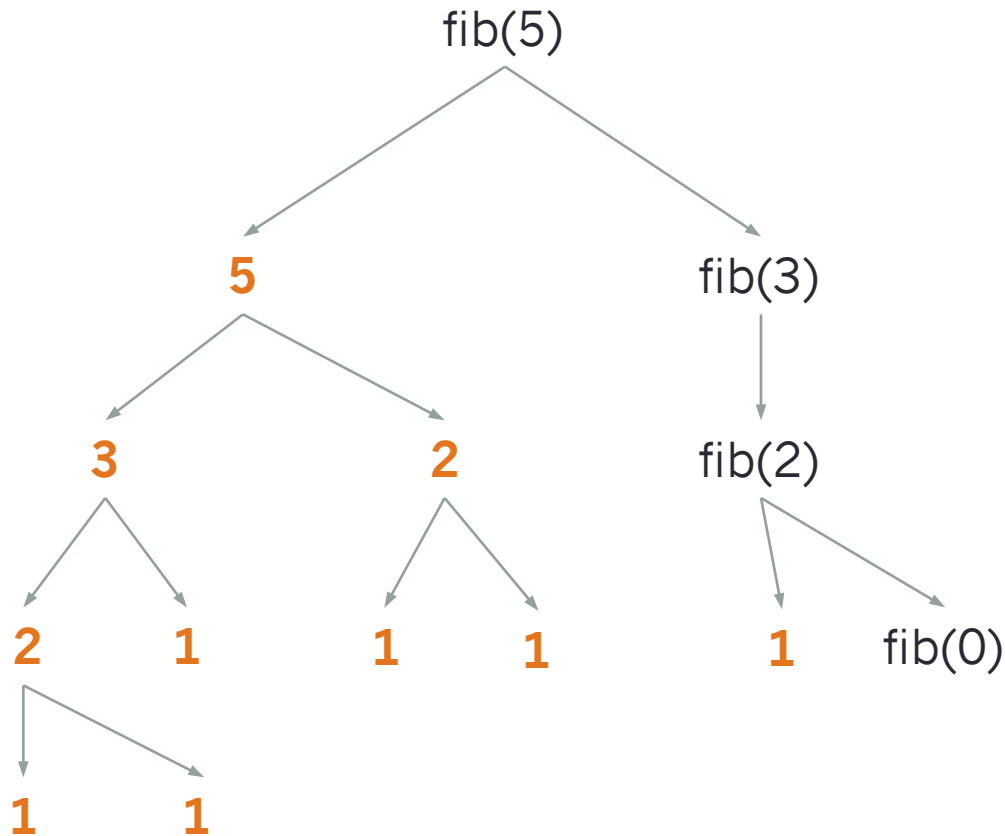
Fibonacci Function Stack



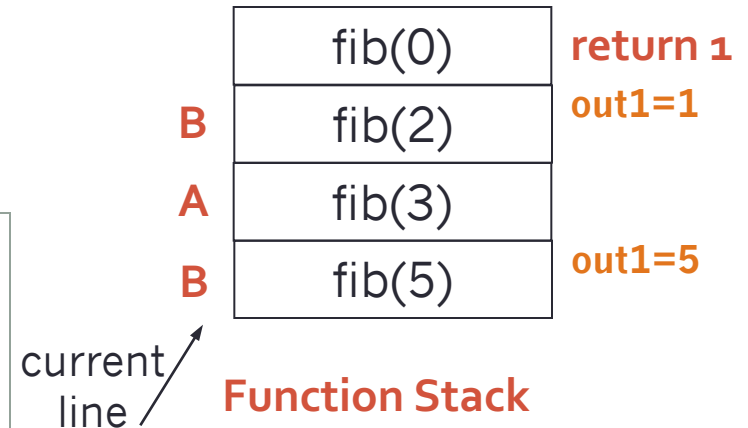
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



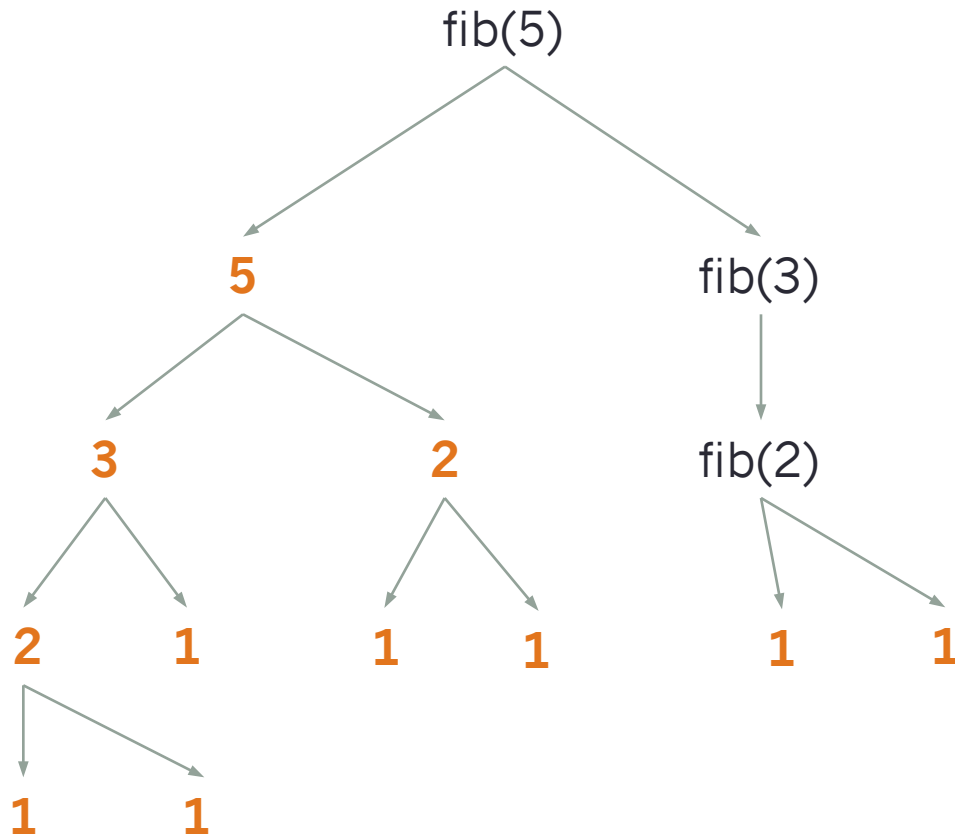
Fibonacci Function Stack



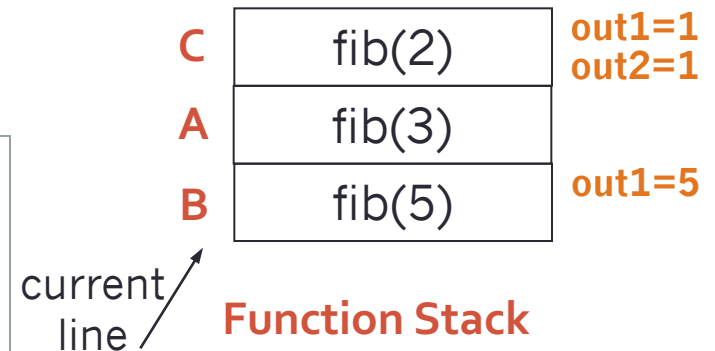
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



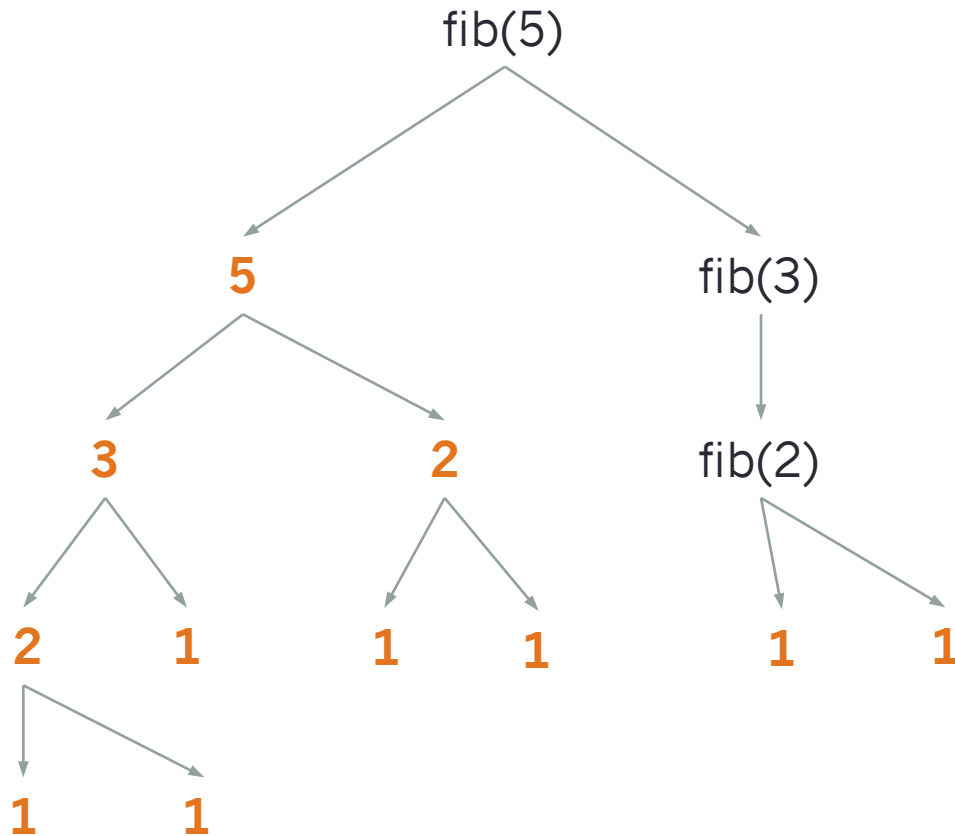
Fibonacci Function Stack



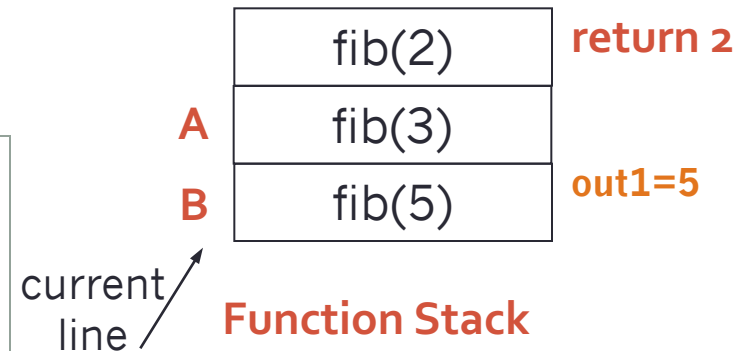
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



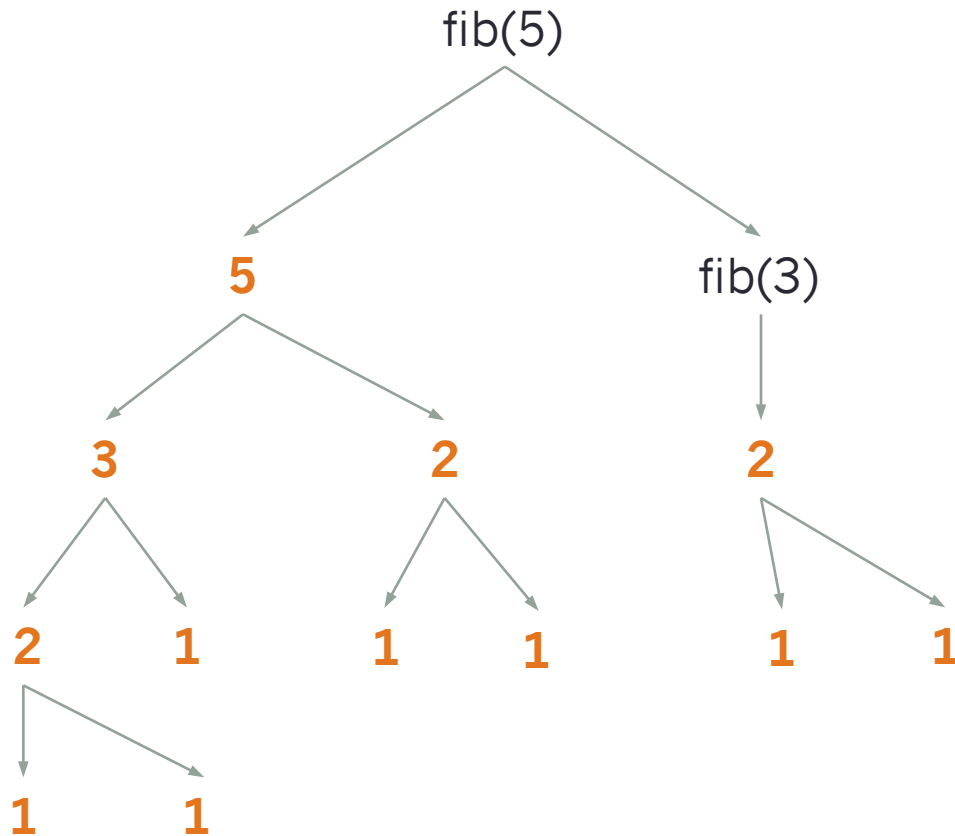
Fibonacci Function Stack



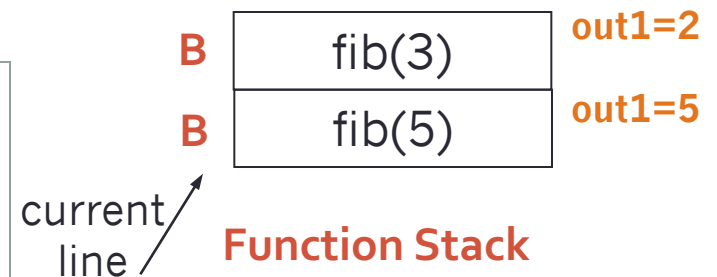
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



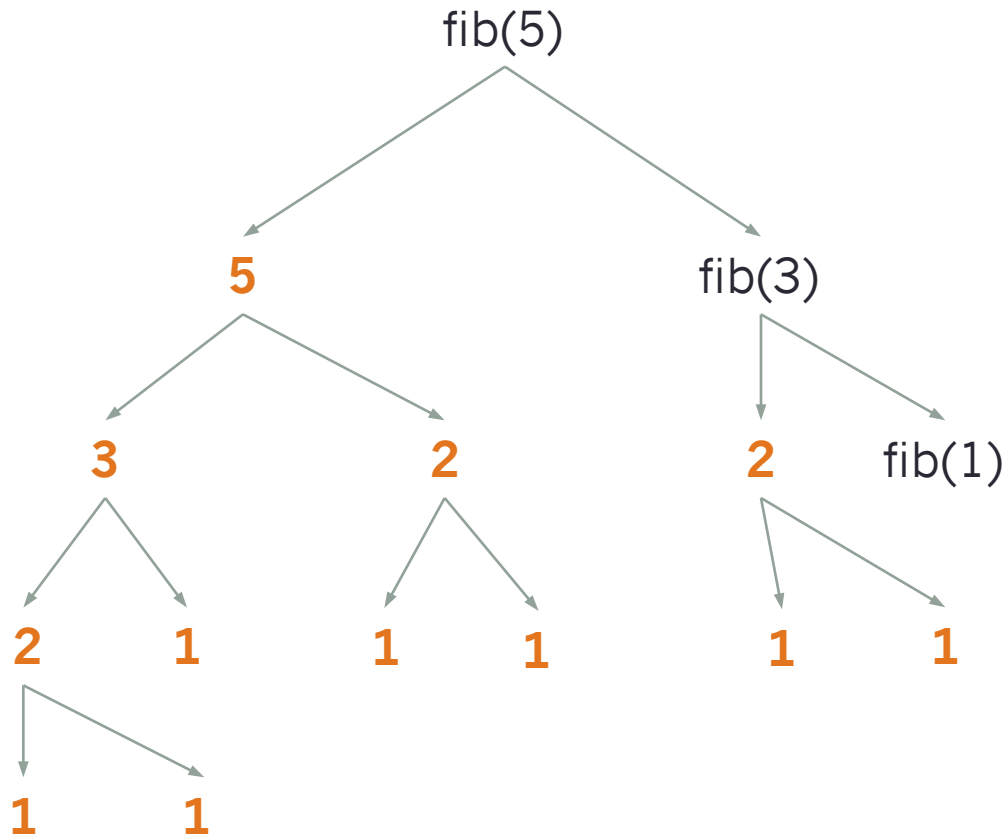
Fibonacci Function Stack



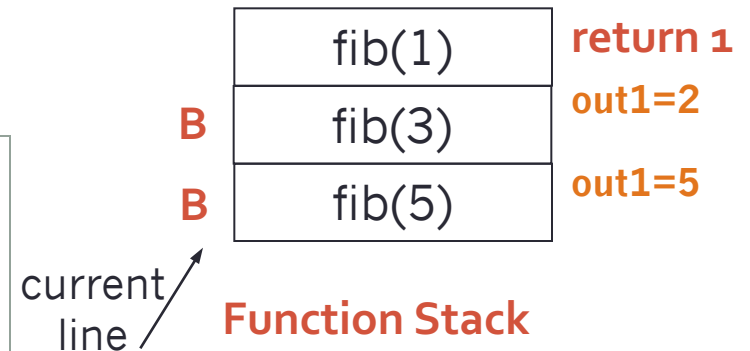
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



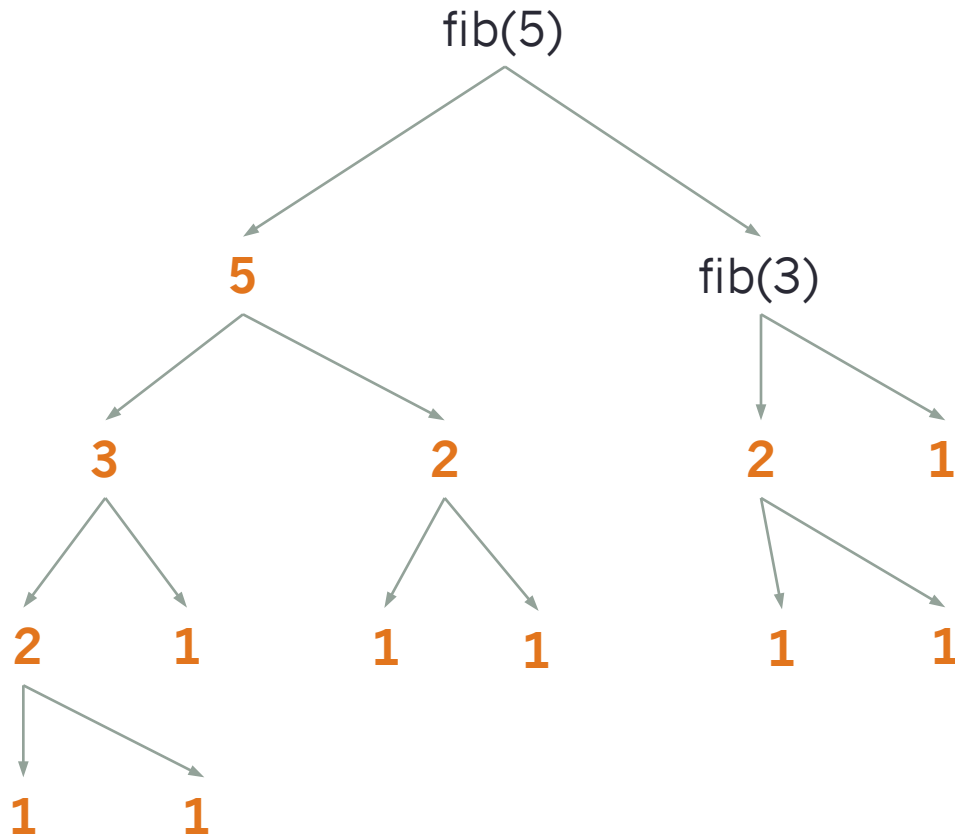
Fibonacci Function Stack



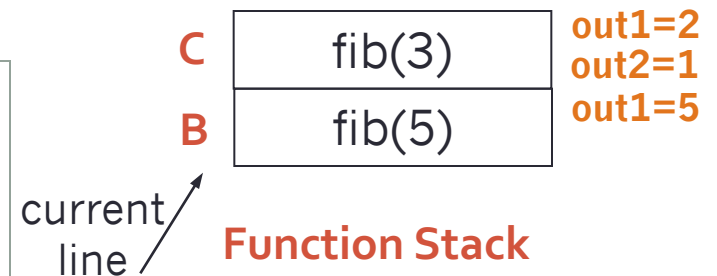
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



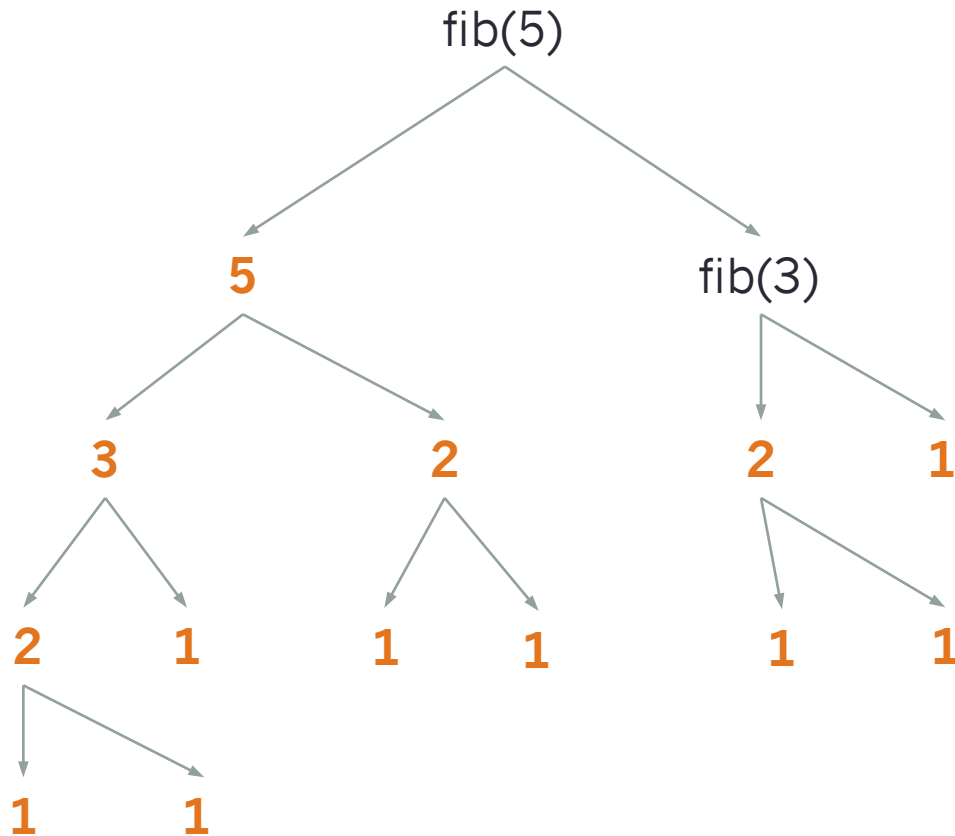
Fibonacci Function Stack



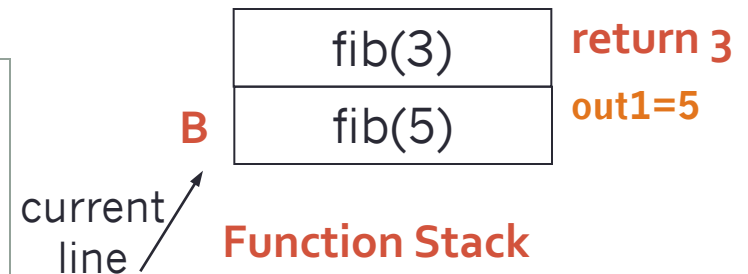
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



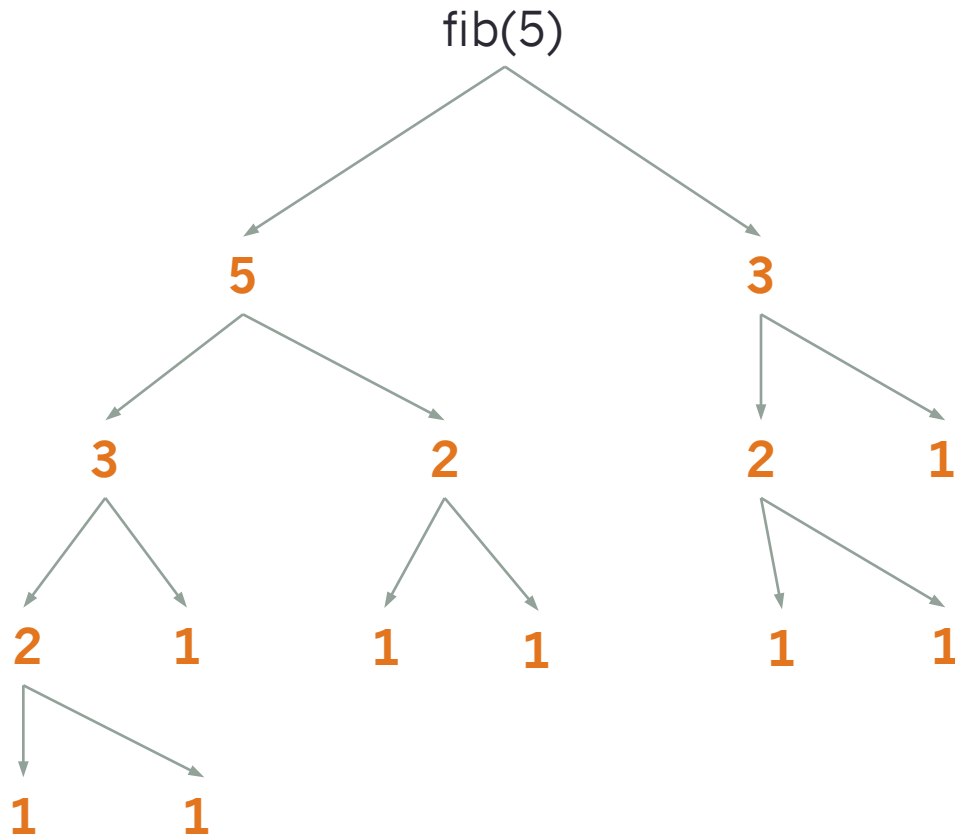
Fibonacci Function Stack



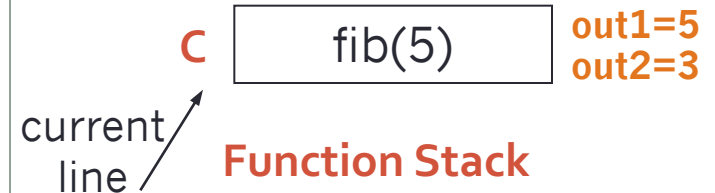
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



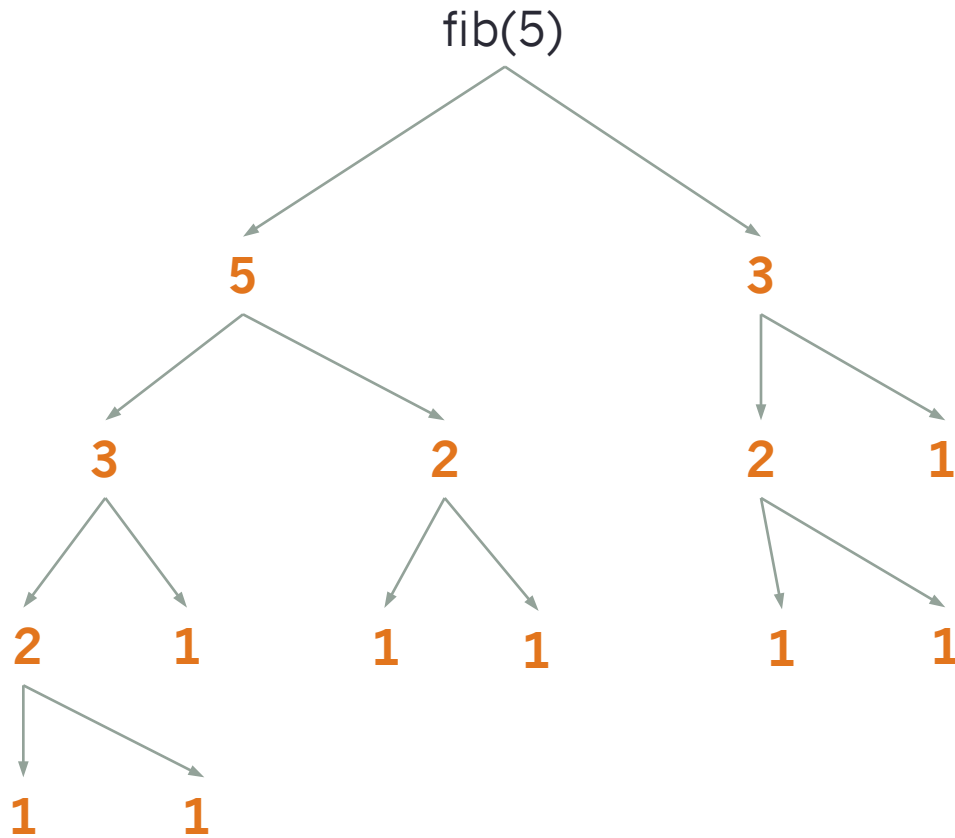
Fibonacci Function Stack



```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```



Fibonacci Function Stack

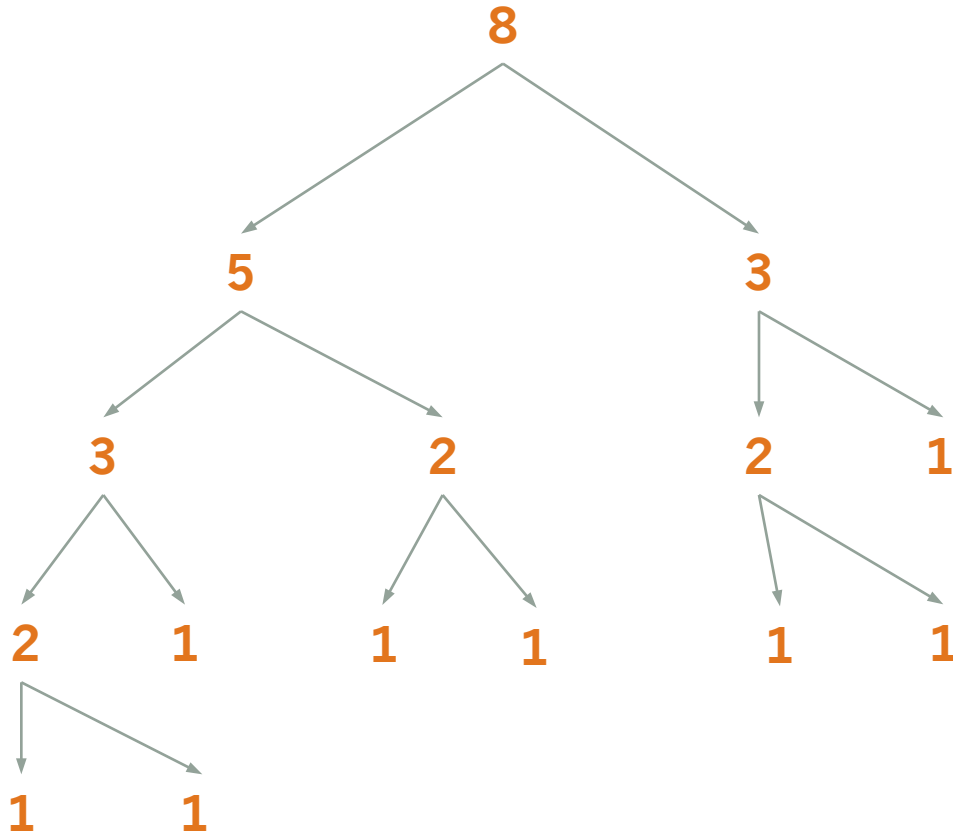


```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```

current
line ↗

fib(5) return 8
Function Stack

Fibonacci Function Stack

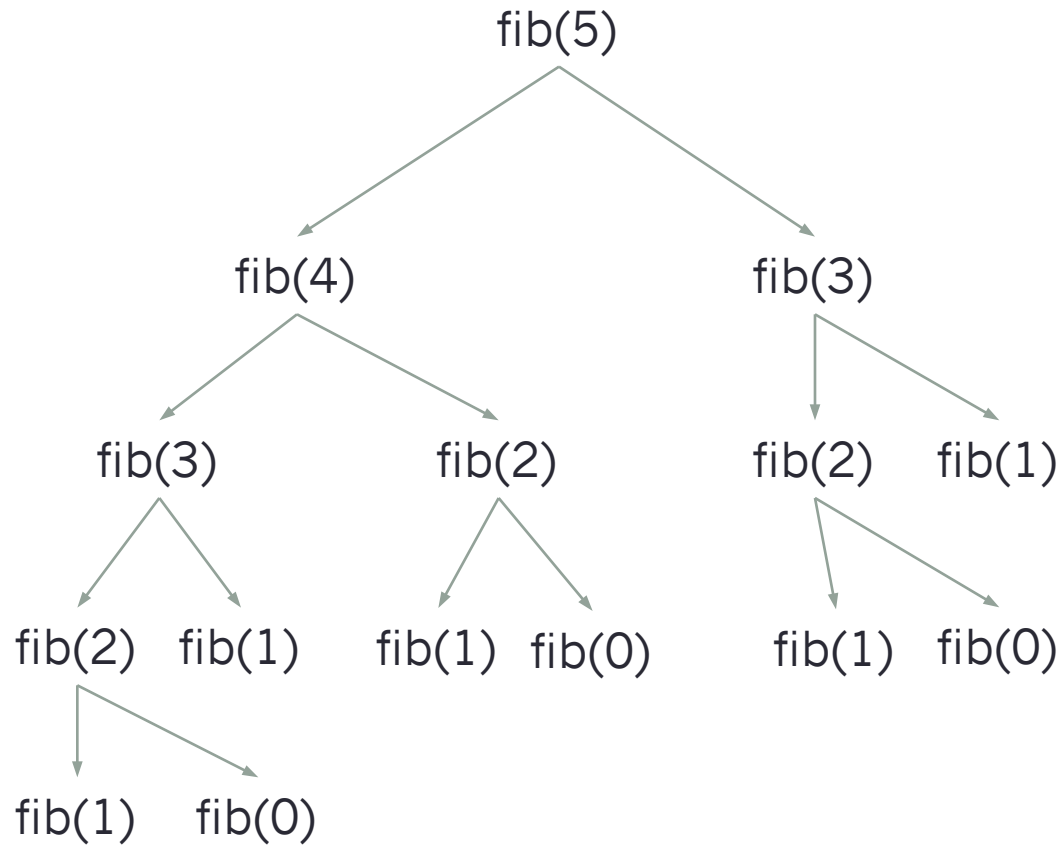


```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```

empty!

Function Stack

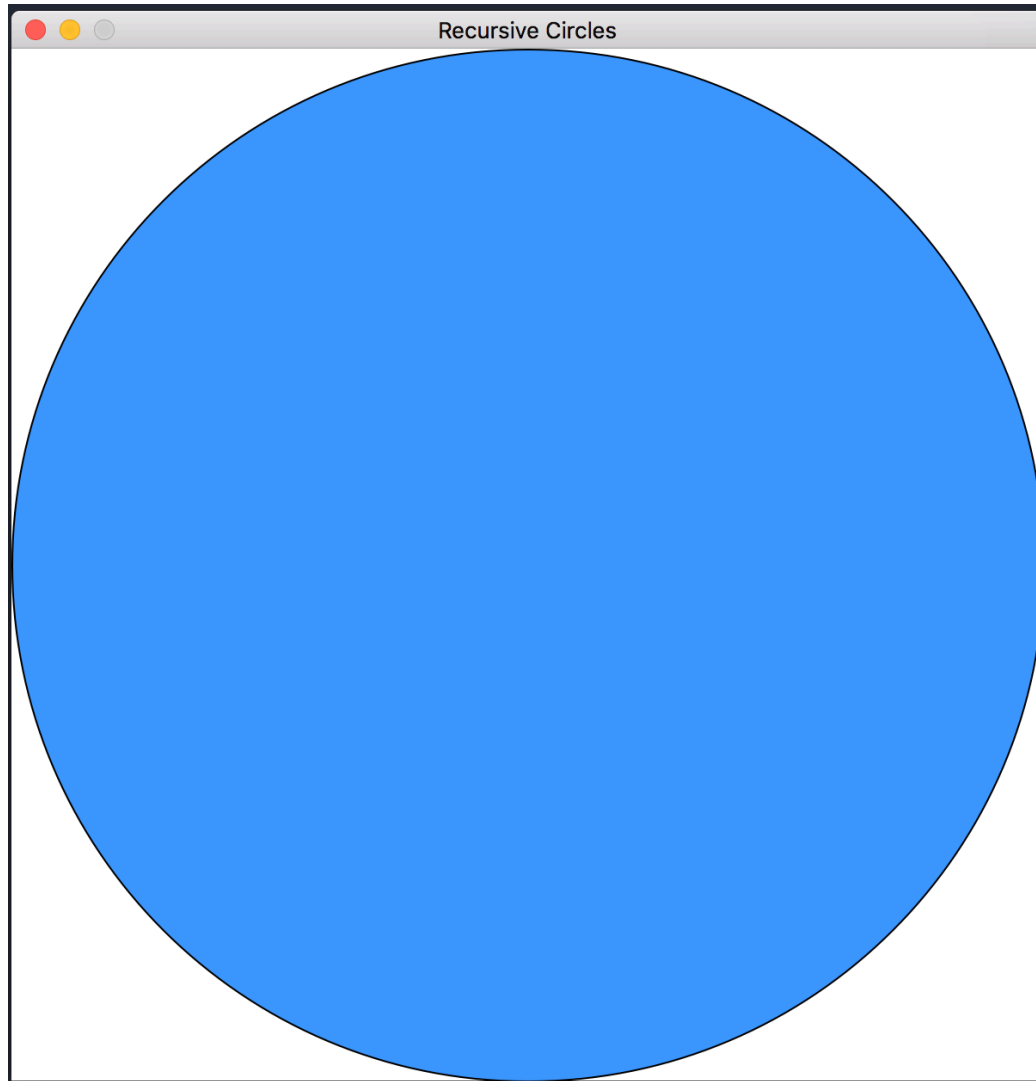
Fibonacci Tree with Function Calls



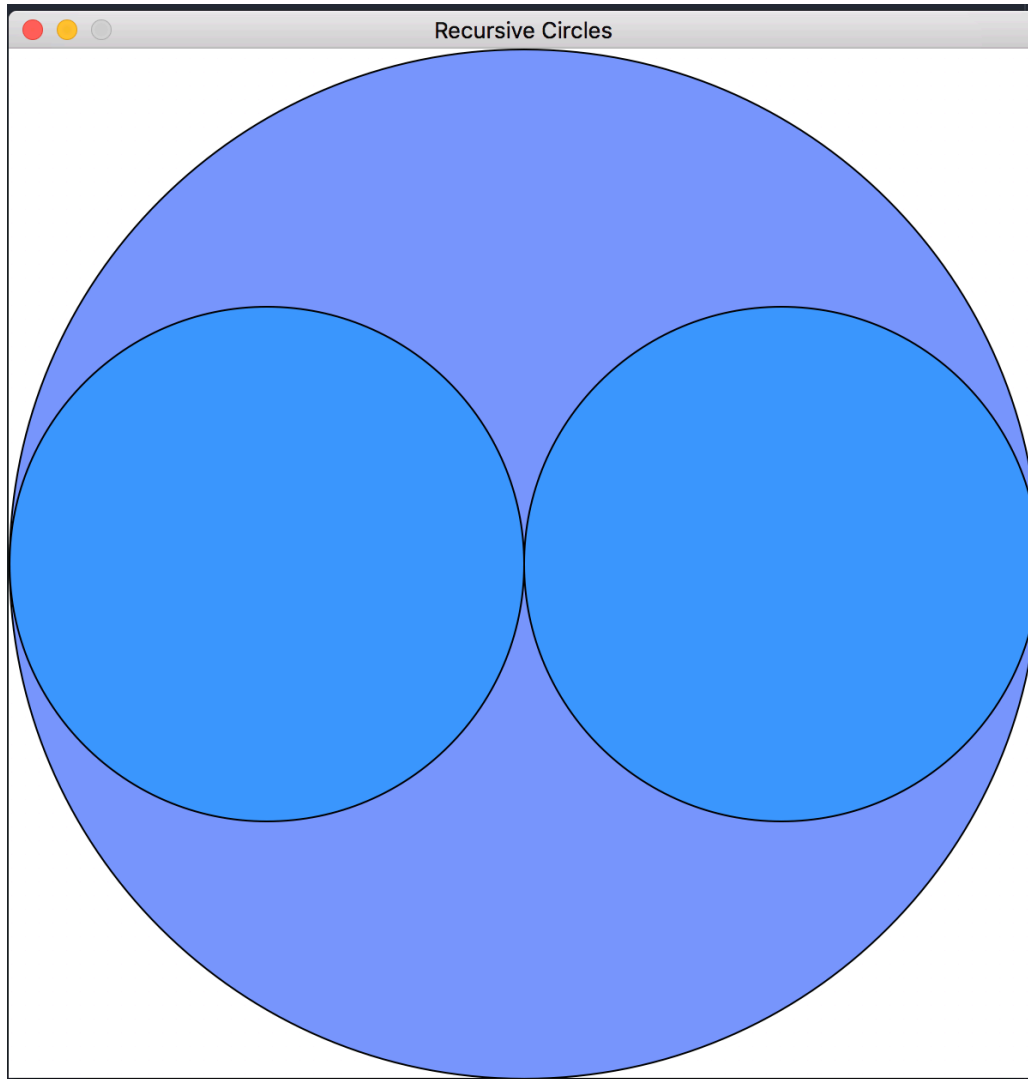
```
def fib(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        Line A → out1 = fib(n-1)  
        Line B → out2 = fib(n-2)  
        Line C → return out1 + out2
```

Recursive Circles Example

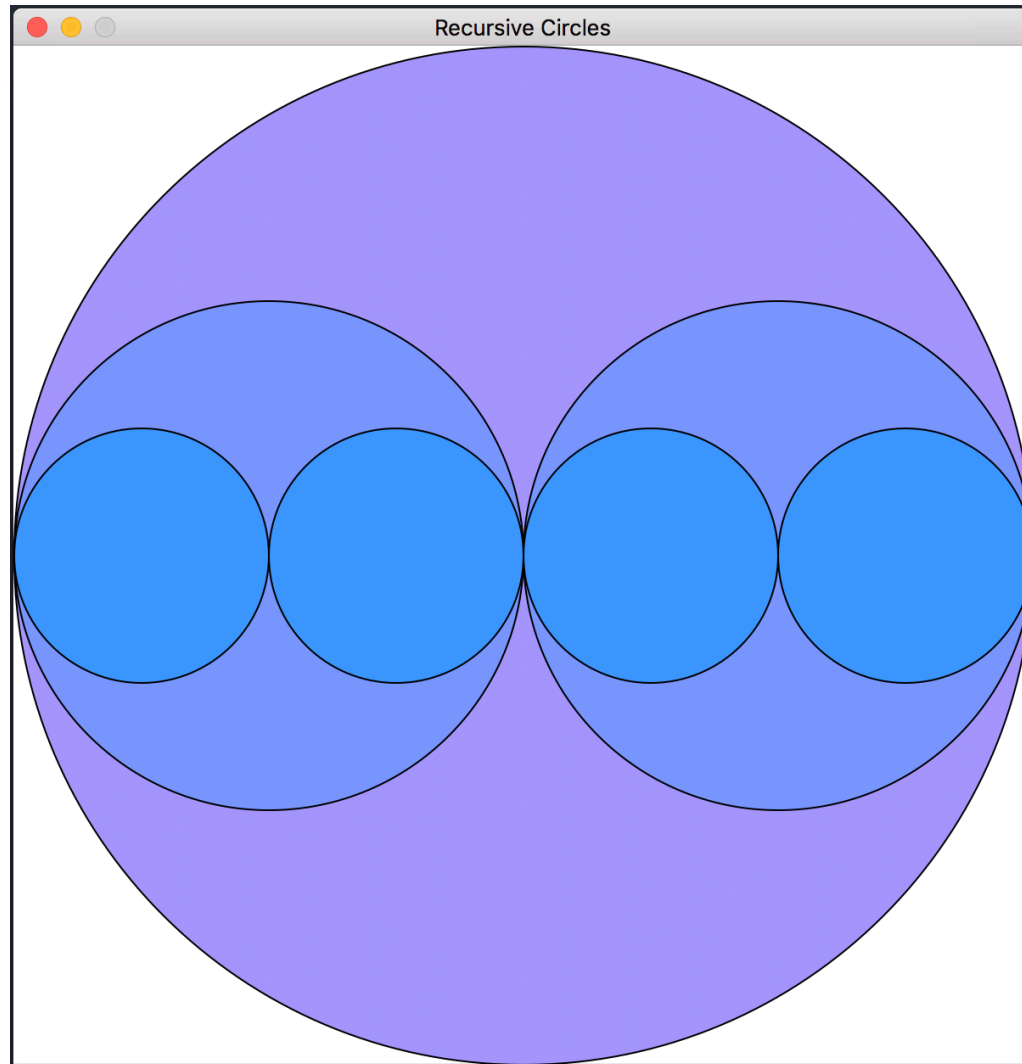
Recursive circles, $n=1$



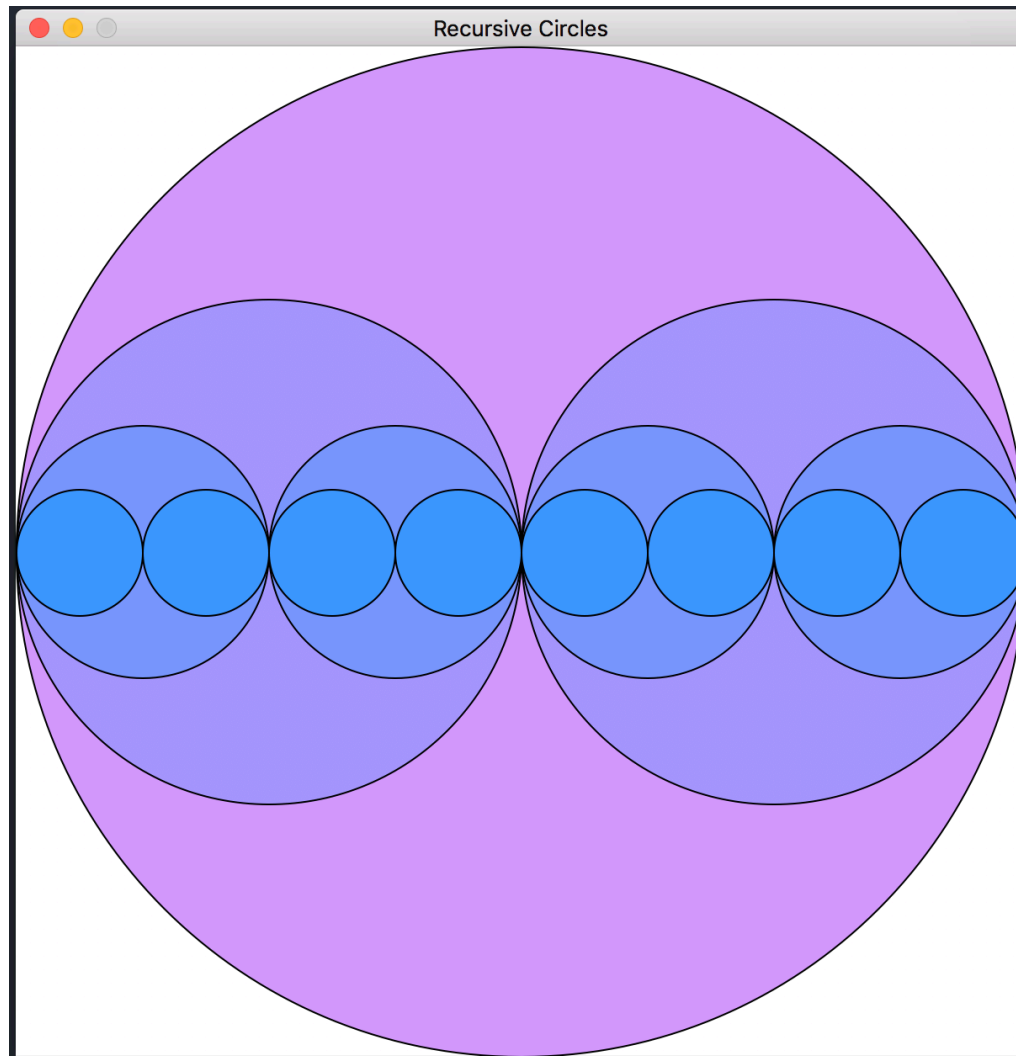
Recursive circles, $n=2$



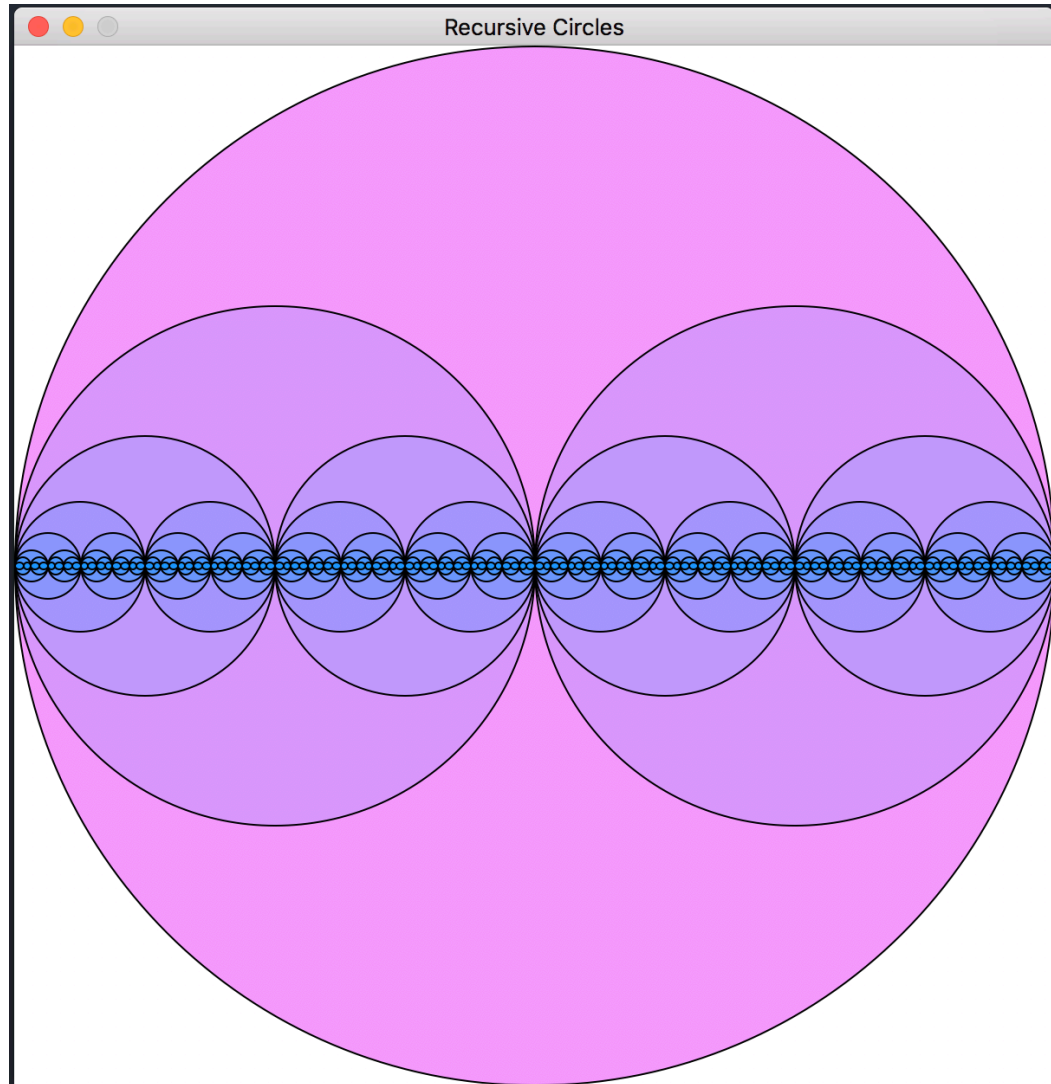
Recursive circles, $n=3$



Recursive circles, $n=4$



Recursive circles, $n=8$



Recap Quiz 5

- ②
- (a) $3n$ steps $\rightarrow O(n)$
 - (b) 100 steps $\rightarrow O(1)$
 - (c) $n \cdot \log n \rightarrow O(n \log n)$
 - (d) $n \cdot \frac{n}{2} \rightarrow O(n^2)$

③ not sorted!

$$n^2 = 10$$

$$n = \sqrt{10}$$

$$O(n^2)$$

$$(2n)^2$$

$$4n^2$$

$$(2\sqrt{10})^2 = 40$$

⑤

0	1	2	3	4	5
s	o	r	t	e	d
d	o	r	t	e	s
d	e	r	t	o	s

swap(0, 5, lst)
swap(1, 4, lst)

⑥

$lst[j] < lst[j+1]$

swap(j, j+1, lst)

5, 3, 8, 4, 2

↑
place min