

# Solution Review : Compute nth Fibonacci Number

This lesson will explain how to compute the nth Fibonacci number using recursion.

## WE'LL COVER THE FOLLOWING ^

- Solution: Use Recursion

## Solution: Use Recursion #

The Fibonacci sequence is obtained by adding the previous two consecutive terms; they are defined by the sequence,

```
fibonacci(0) = 0 # base case
fibonacci(1) = 1

fibonacci(n) = fibonacci(n - 1) + fibonacci(n - 2) for n >= 2 # recursive case
```

For example ,

```
0,1,1,2,3,5,8,.....
```

```
if n=2, f(n)=3
```

The following illustration explains the concept by calculating the **fourth** Fibonacci number.

fibonacci(4)

Recursion tree of fibonacci(4)

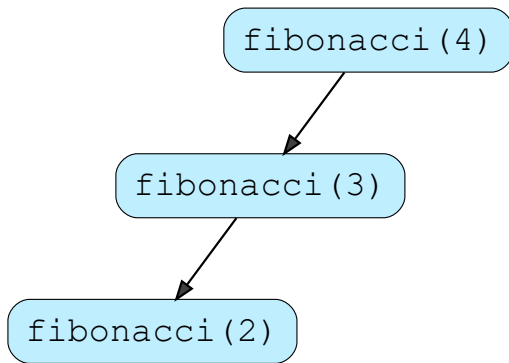
1 of 14

fibonacci(4)

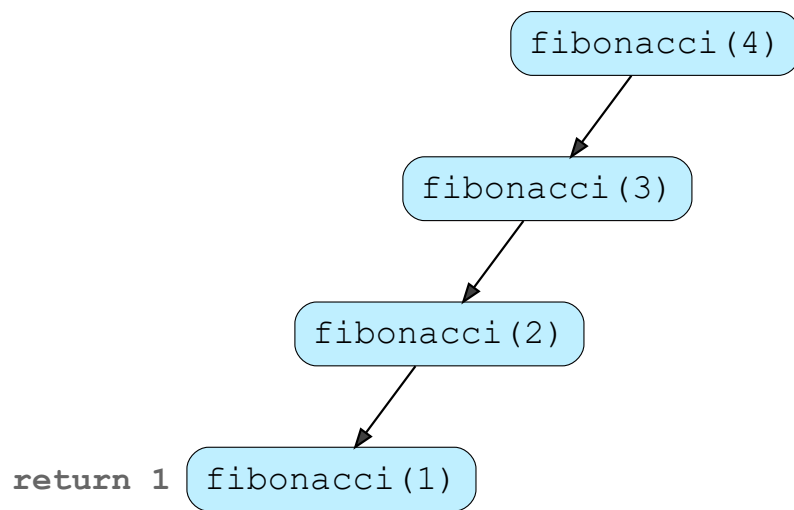


fibonacci(3)

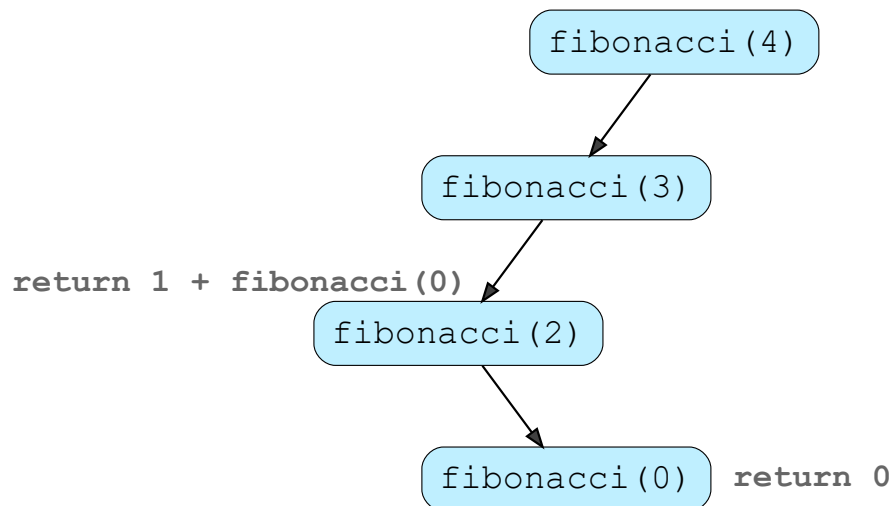
2 of 14



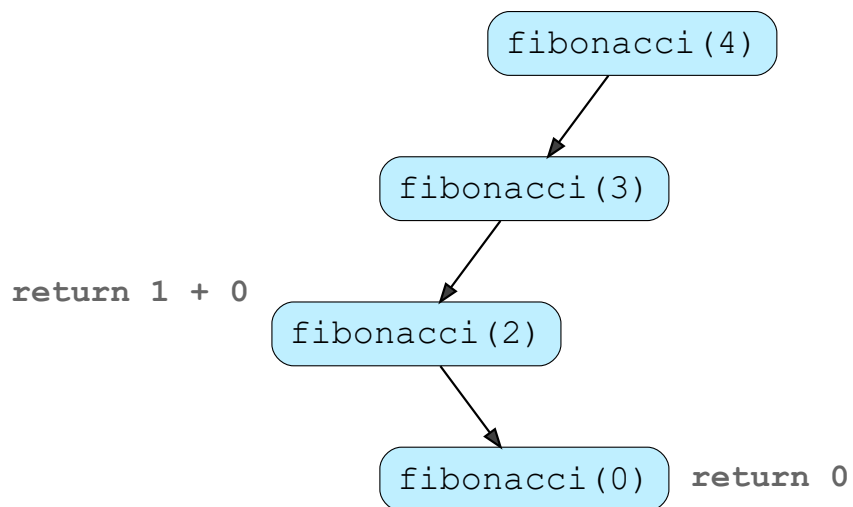
3 of 14



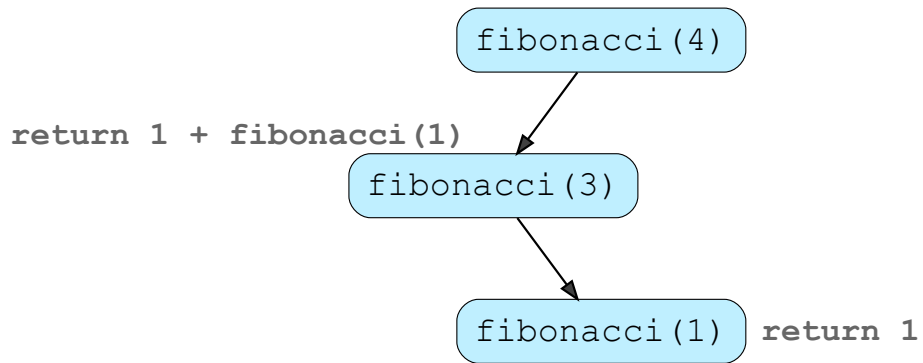
4 of 14



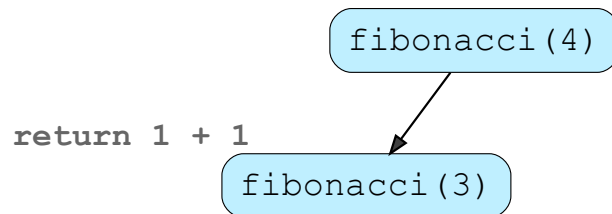
5 of 14



6 of 14



7 of 14



8 of 14

```
return 2 + fibonacci(2)
```

fibonacci(4)



fibonacci(2)

9 of 14

```
return 2 + fibonacci(2)
```

fibonacci(4)



fibonacci(2)



```
return 1
```

fibonacci(1)

10 of 14

```
return 2 + fibonacci(2)
```

fibonacci(4)

```
return 1 + fibonacci(0)
```

fibonacci(2)

fibonacci(0) return 1

11 of 14

```
return 2 + fibonacci(2)
```

fibonacci(4)

```
return 1 + 0
```

fibonacci(2)

12 of 14

```
return 2 + 1
```

```
fibonacci(4)
```

13 of 14

**fibonacci(4) = 3**

14 of 14

—

[ ]

The following python code demonstrates how to find the nth fibonacci number:

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





```
print(fibonacci(4))
```



Let's move on to the next problem.