Funktionen und Rekursion

D. Komm, T. Rupf, J. Závodný
05.02.2019

Treppensteigen

Eine Treppe hat 20 Stufen. Du kannst entweder auf jede Stufe einzeln treten, oder auch einmal eine überspringen und damit gleich zwei Stufen auf einmal nehmen. **Wie viele Möglichkeiten gibt es, die Treppe mit den 20 Stufen hochzugehen?**

Beispiel: Du kannst 4 Stufen auf 5 Arten hochgehen:

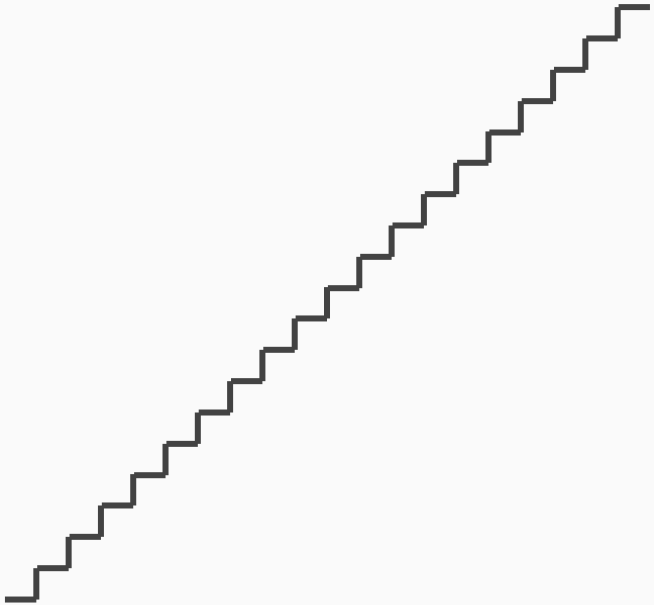
1-1-1-1, 1-1-2, 1-2-1, 2-1-1, 2-2

Allgemeiner Problemlösungstipp

Allgemeiner Problemlösungstipp

Zeichne ein Bild!

Treppensteigen

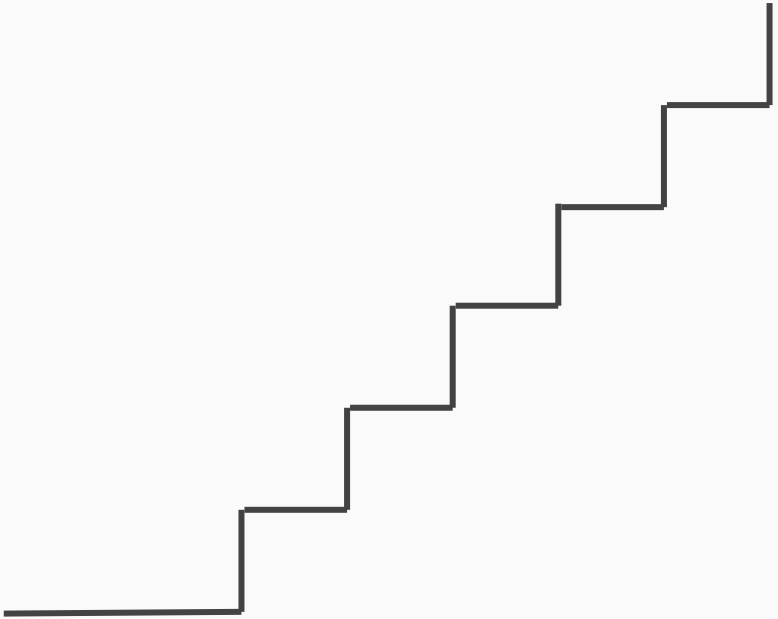


2. Allgemeiner Problemlösungstipp

2. Allgemeiner Problemlösungstipp

Beginne mit einem kleinen Beispiel!

Treppensteigen

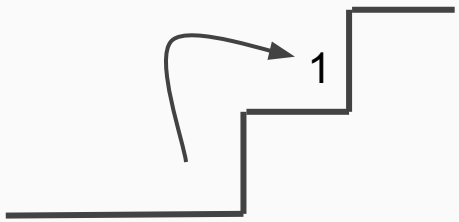


2. Allgemeiner Problemlösungstipp

Beginne mit einem kleinen Beispiel!

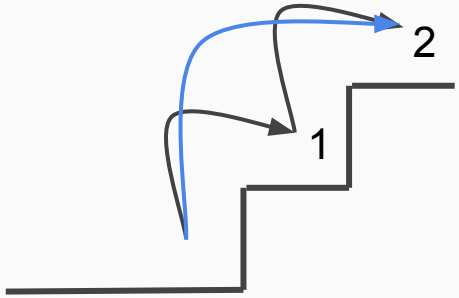
Treppensteigen

Treppe mit 1 Stufe: 1 Möglichkeit



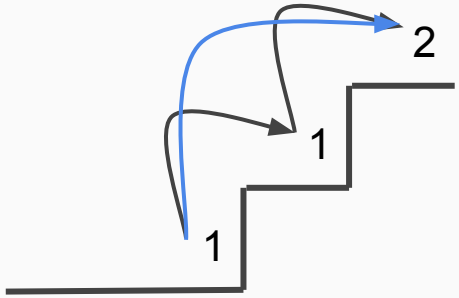
Treppensteigen

Treppe mit 1 Stufe: 1 Möglichkeit
Treppe mit 2 Stufen: 2 Möglichkeiten



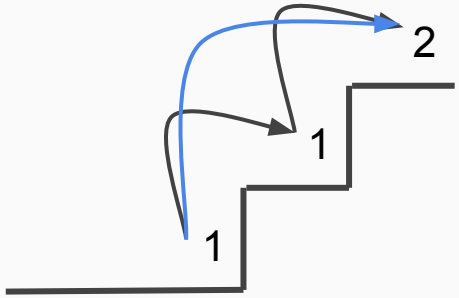
Treppensteigen

Treppe mit 0 Stufen: 1 Möglichkeit
Treppe mit 1 Stufe: 1 Möglichkeit
Treppe mit 2 Stufen: 2 Möglichkeiten



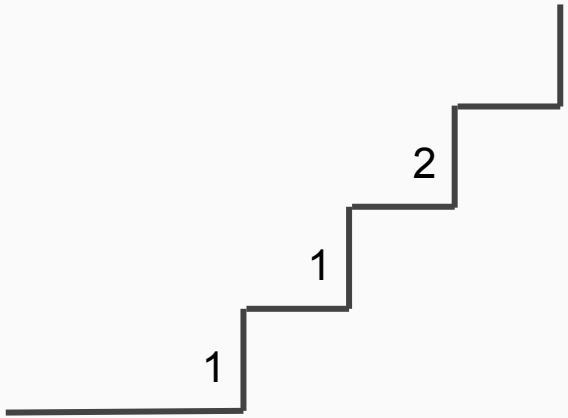
Treppensteigen

m(0) = 1
m(1) = 1
m(2) = 2



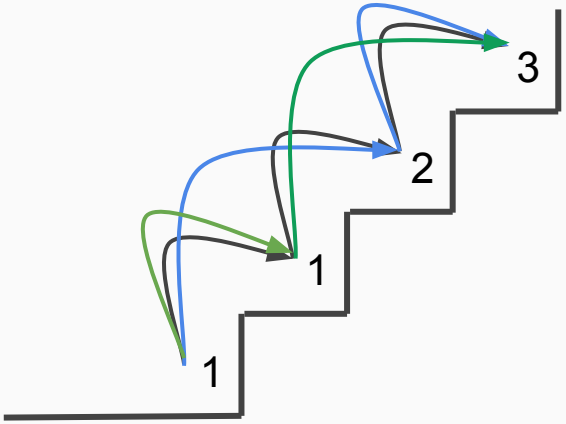
Treppensteigen

m(0) = 1
m(1) = 1
m(2) = 2
m(3) = ?



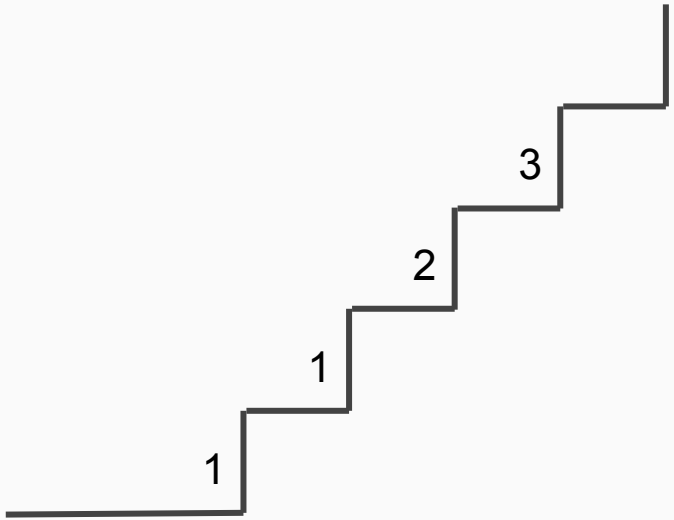
Treppensteigen

m(0) = 1
m(1) = 1
m(2) = 2
m(3) = 3



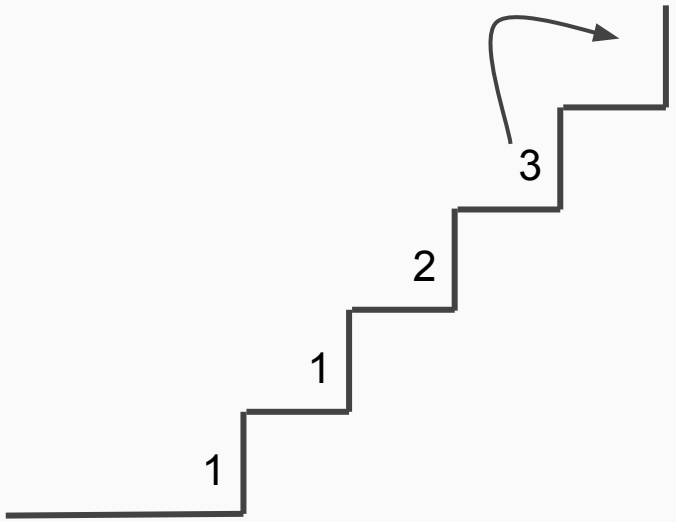
Treppensteigen

m(0) = 1
m(1) = 1
m(2) = 2
m(3) = 3
m(4) = ?



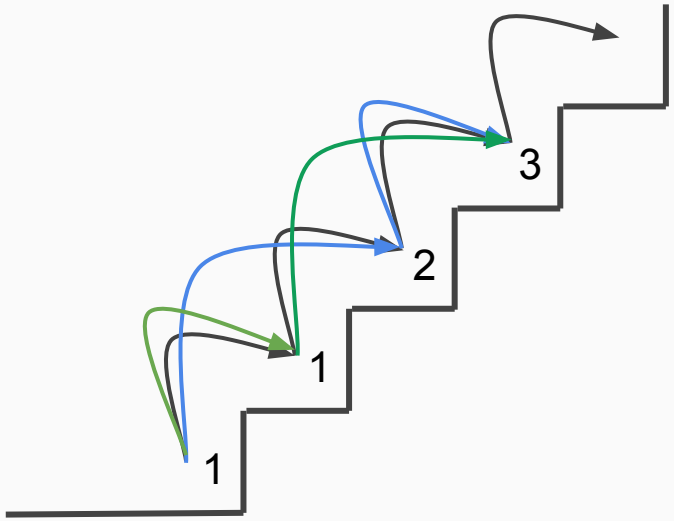
Treppensteigen

m(0) = 1
m(1) = 1
m(2) = 2
m(3) = 3
m(4) = ?



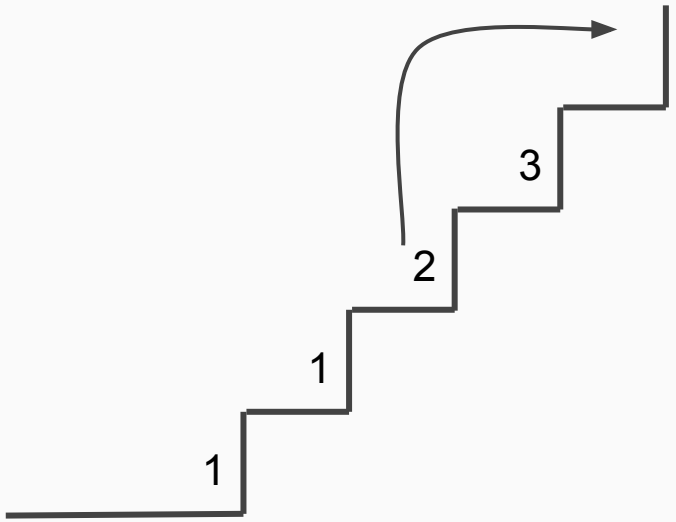
Treppensteigen

m(0) = 1
m(1) = 1
m(2) = 2
m(3) = 3
m(4) = ?



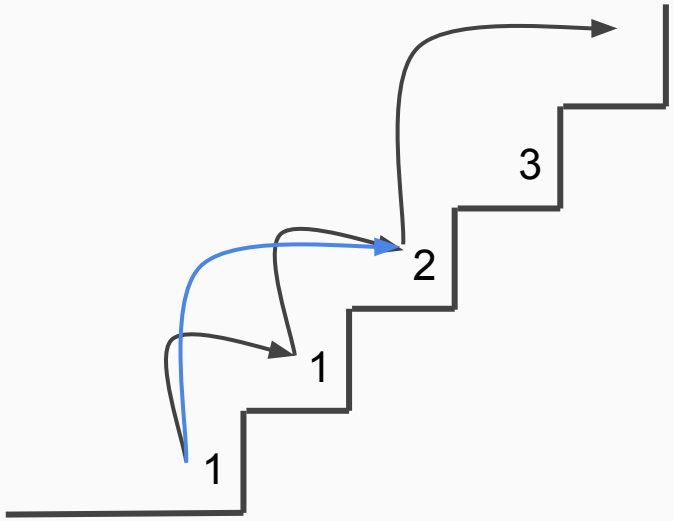
Treppensteigen

m(0) = 1
m(1) = 1
m(2) = 2
m(3) = 3
m(4) = ?



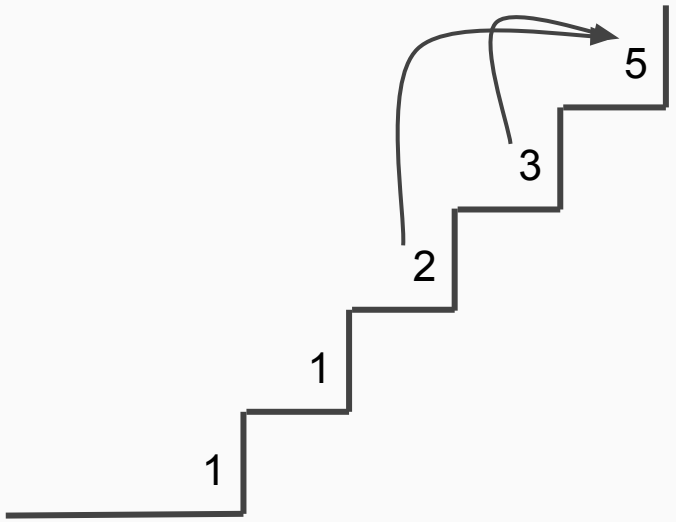
Treppensteigen

m(0) = 1
m(1) = 1
m(2) = 2
m(3) = 3
m(4) = ?



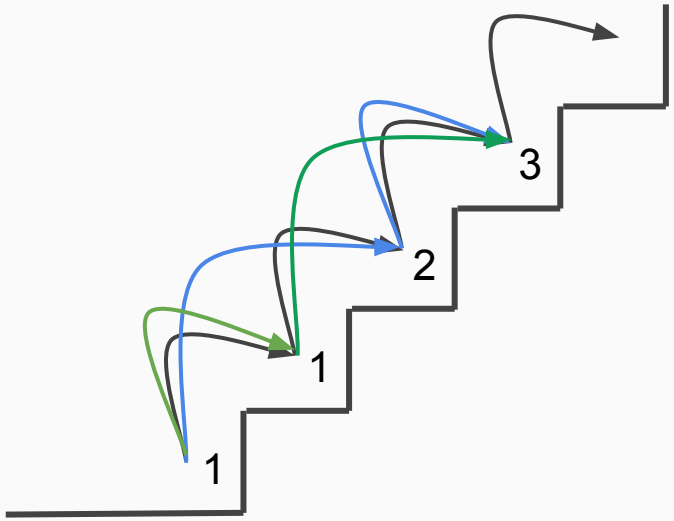
Treppensteigen

m(0) = 1
m(1) = 1
m(2) = 2
m(3) = 3
m(4) = 2 + 3



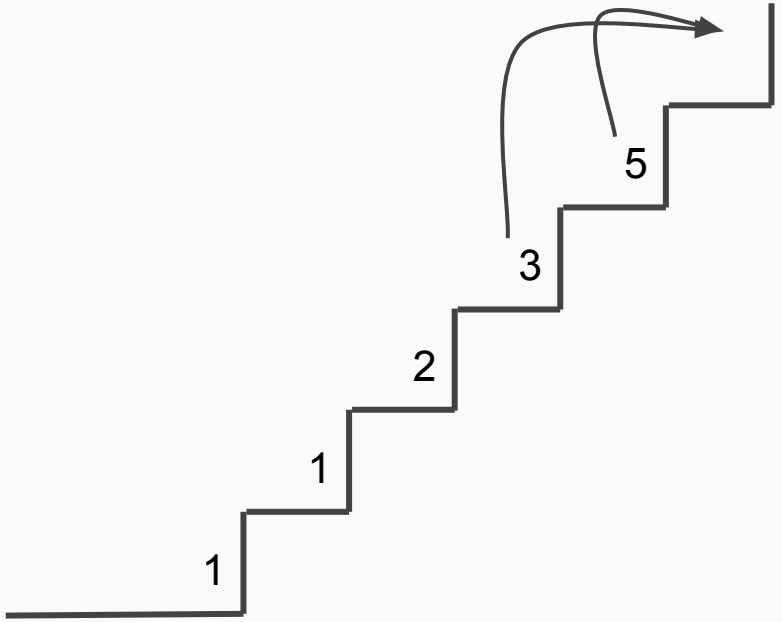
Treppensteigen

m(0) = 1
m(1) = 1
m(2) = 2
m(3) = 3
m(4) = ?



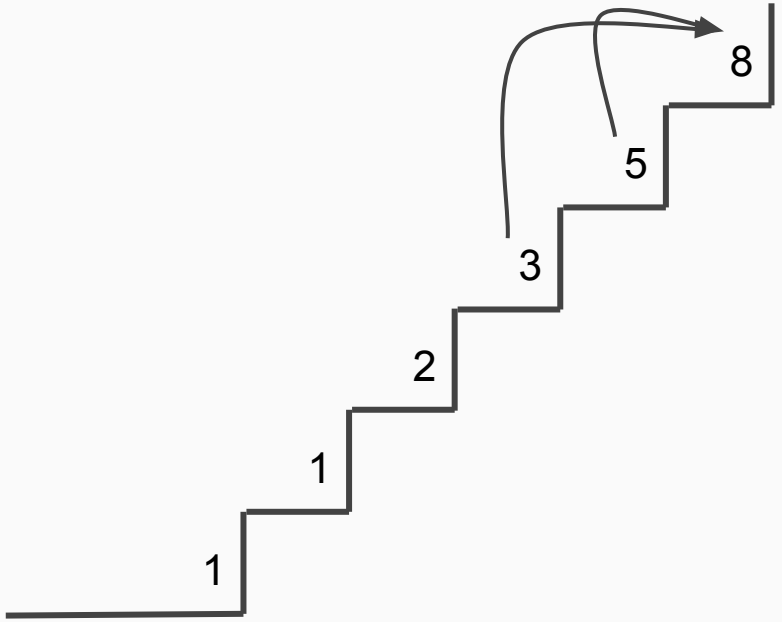
Treppensteigen

m(0) = 1
m(1) = 1
m(2) = 2
m(3) = 3
m(4) = 5
m(5) =



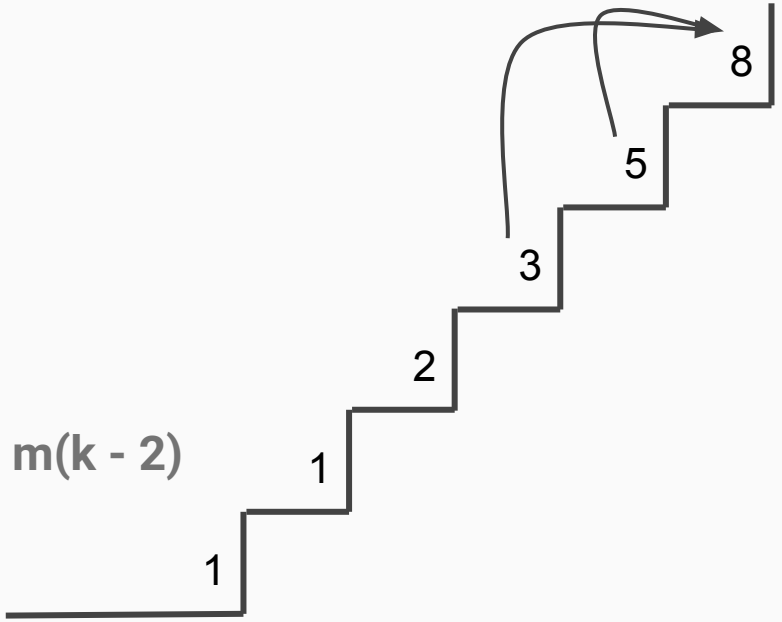
Treppensteigen

m(0) = 1
m(1) = 1
m(2) = 2
m(3) = 3
m(4) = 5
m(5) = 8



Treppensteigen

m(0) = 1
m(1) = 1
m(2) = 2
m(3) = 3
m(4) = 5
m(5) = 8
m(k) = m(k - 1) + m(k - 2)



Treppensteigen = Fibonacci-Zahlen

Fibonacci-Zahlen: $F(0) = 1, \quad F(1) = 1, \quad F(k) = F(k - 1) + F(k + 2)$

Treppensteigen = Fibonacci-Zahlen

Fibonacci-Zahlen: $F(0) = 1, \quad F(1) = 1, \quad F(k) = F(k - 1) + F(k + 2)$

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

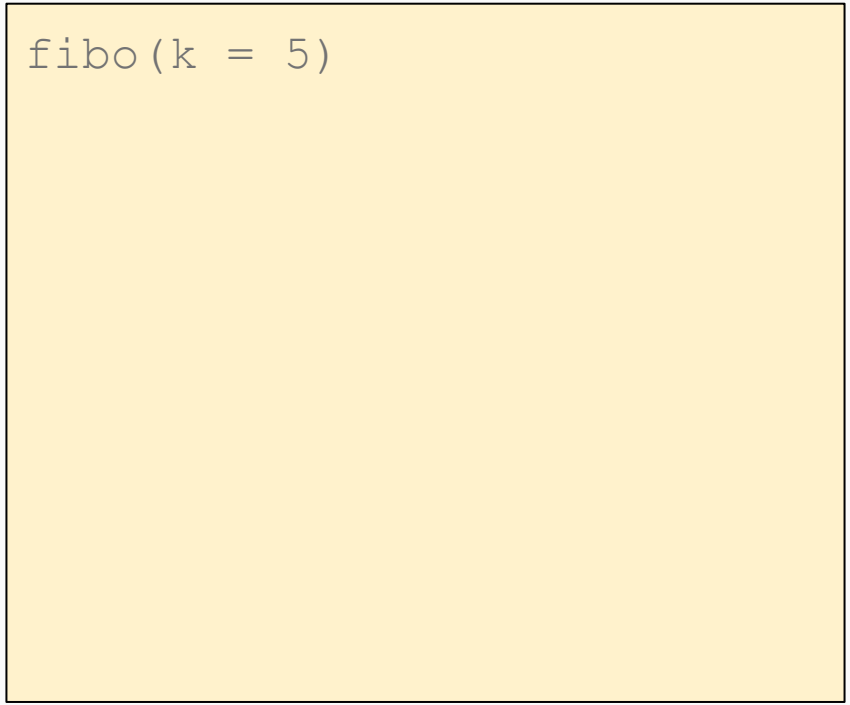
Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```




```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
```

```
fibo(k = 3)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
```

```
fibo(k = 3)
  return fibo(2) + fibo(1)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
```

```
fibo(k = 3)
  return fibo(2) +
```

```
fibo(k = 2)
  return fibo(1) + fibo(0)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
```

```
fibo(k = 3)
  return fibo(2) +
```

```
fibo(k = 2)
  return fibo(1) + fibo(0)
```

```
fibo(k = 1)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
```

```
fibo(k = 3)
  return fibo(2) +
```

```
fibo(k = 2)
  return fibo(1) + fibo(0)
```

```
fibo(k = 1)
  return 1
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
```

```
fibo(k = 3)
  return fibo(2) +
```

```
fibo(k = 2)
  return fibo(1) + fibo(0)
           1
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
```

```
fibo(k = 3)
  return fibo(2) +
```

```
fibo(k = 2)
  return fibo(1) + fibo(0)
```

```
fibo(k = 0)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
```

```
fibo(k = 3)
  return fibo(2) +
```

```
fibo(k = 2)
  return fibo(1) + fibo(0)
```

```
fibo(k = 0)
  return 1
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
```

```
fibo(k = 3)
  return fibo(2) +
```

```
fibo(k = 2)
  return fibo(1) + fibo(0)
           1          1
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
```

```
fibo(k = 3)
  return fibo(2) +
```

```
fibo(k = 2)
  return 2
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
```

```
fibo(k = 3)
  return fibo(2) + fibo(1)
                   2
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
```

```
fibo(k = 3)
  return fibo(2) + fibo(1)
                   2
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
```

```
fibo(k = 3)
  return fibo(2) + fibo(1)
                   2
```

```
fibo(k = 1)
  return 1
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
```

```
fibo(k = 3)
  return fibo(2) + fibo(1)
                   2         1
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
```

```
fibo(k = 3)
  return 2 + 1
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
```

```
3
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
           3
```

```
fibo(k = 2)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
                   3
```

```
fibo(k = 2)
  return fibo(1) + fibo(0)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
           3
```

```
fibo(k = 2)
  return fibo(1) + fibo(0)
```

```
fibo(k = 1)
  return 1
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
                   3
```

```
fibo(k = 2)
  return fibo(1) + fibo(0)
                   1
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
                   3
```

```
fibo(k = 2)
  return fibo(1) + fibo(0)
```

```
fibo(k = 0)
  return 1
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
                   3
```

```
fibo(k = 2)
  return fibo(1) + fibo(0)
                   1
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
           3
```

```
fibo(k = 2)
  return 1 + 1
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 4)
  return fibo(3) + fibo(2)
```

3 2

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
           5
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 3)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
                   5
```

```
fibo(k = 3)
  return fibo(2) + fibo(1)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
                   5
```

```
fibo(k = 3)
  return fibo(2) + fibo(1)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 3)
  return fibo(2) + fibo(1)
```

```
fibo(k = 2)
  return fibo(1) + fibo(0)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
```

```
fibo(k = 3)
  return fibo(2) + fibo(1)
```

```
fibo(k = 2)
  return fibo(1) +
```

```
fibo(k = 1)
  return 1
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
                   5
```

```
fibo(k = 3)
  return fibo(2) + fibo(1)
```

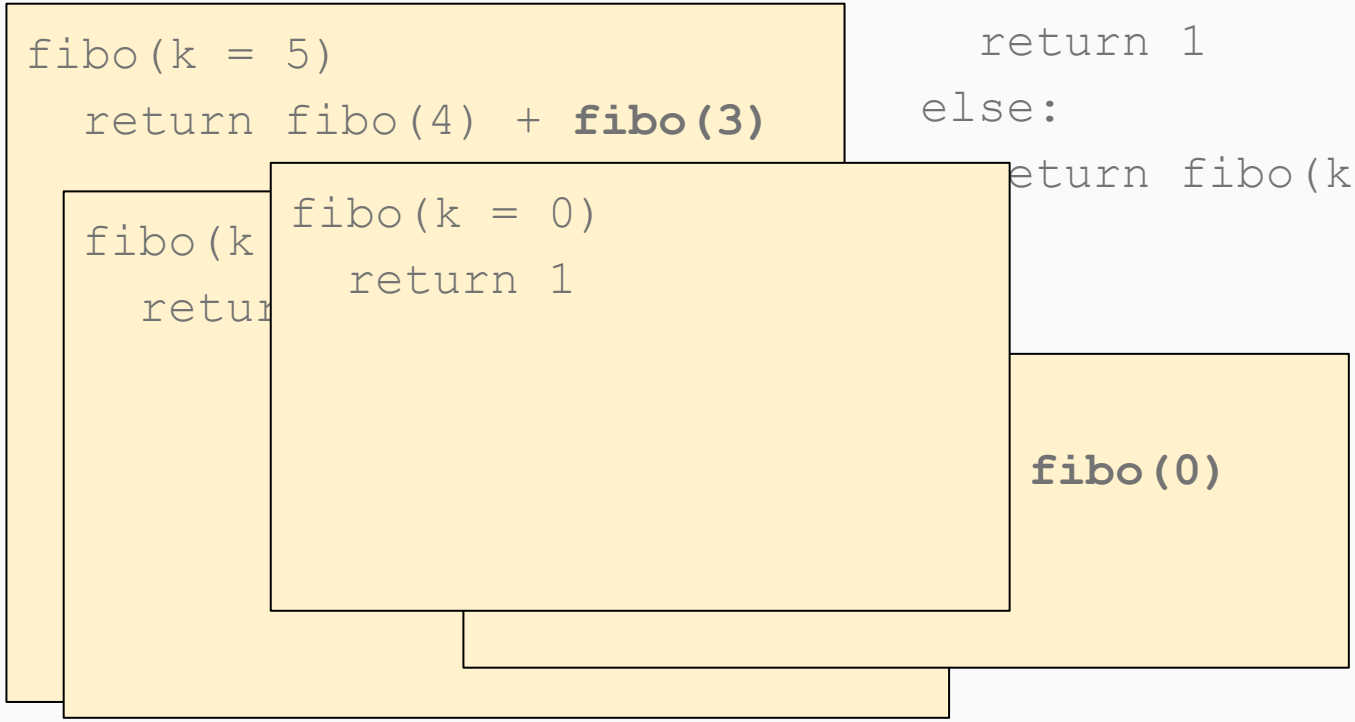
```
fibo(k = 2)
  return fibo(1) + fibo(0)
                   1
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```



Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
                   5
```

```
fibo(k = 3)
  return fibo(2) + fibo(1)
```

```
fibo(k = 2)
  return fibo(1) + fibo(0)
                   1
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
                   5
```

```
fibo(k = 3)
  return fibo(2) + fibo(1)
                   2
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
                   5
```

```
fibo(k = 3)
  return fibo(2) + fibo(1)
                   2
```

```
fibo(k = 1)
  return 1
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
         5
```

```
fibo(k = 3)
  return fibo(2) + fibo(1)
         2         1
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
fibo(k = 5)
  return fibo(4) + fibo(3)
              5          3
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
```

```
fibo(k = 5)
  return 5 + 3
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

```
> fibo(5)
8
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion

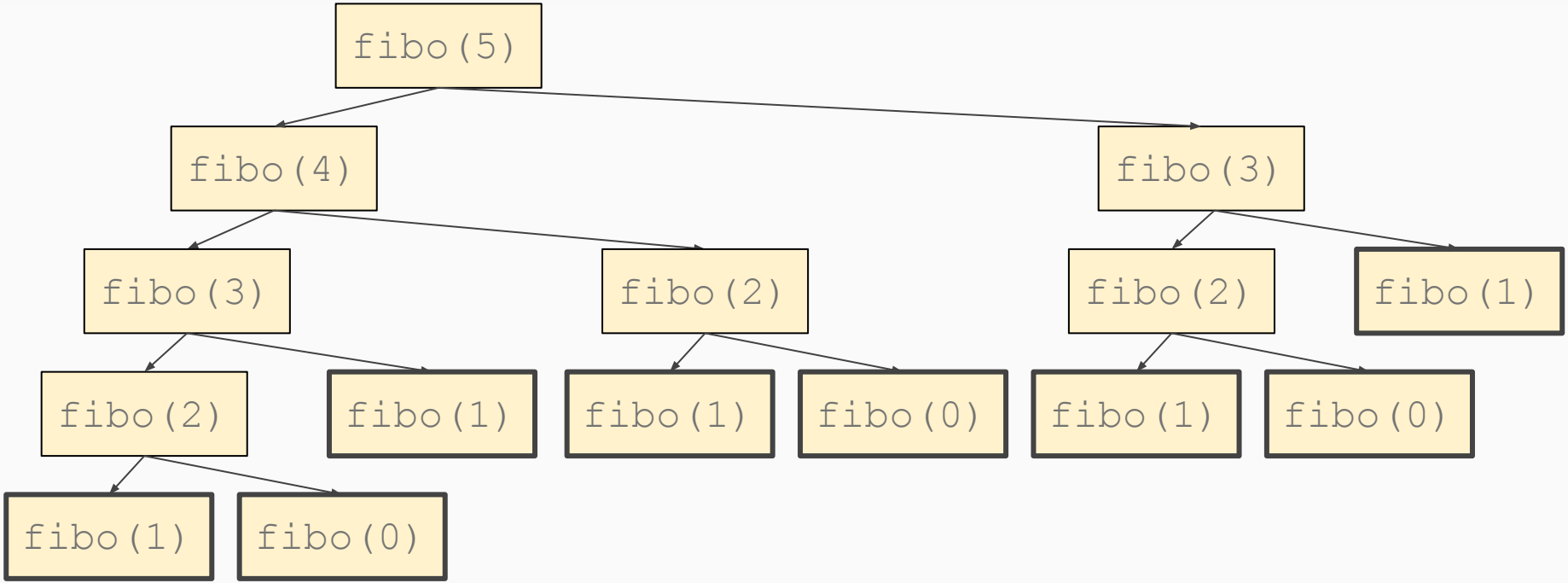
```
> fibo(5)
8
```

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Wie viel Mal wird `fibo()` aufgerufen?

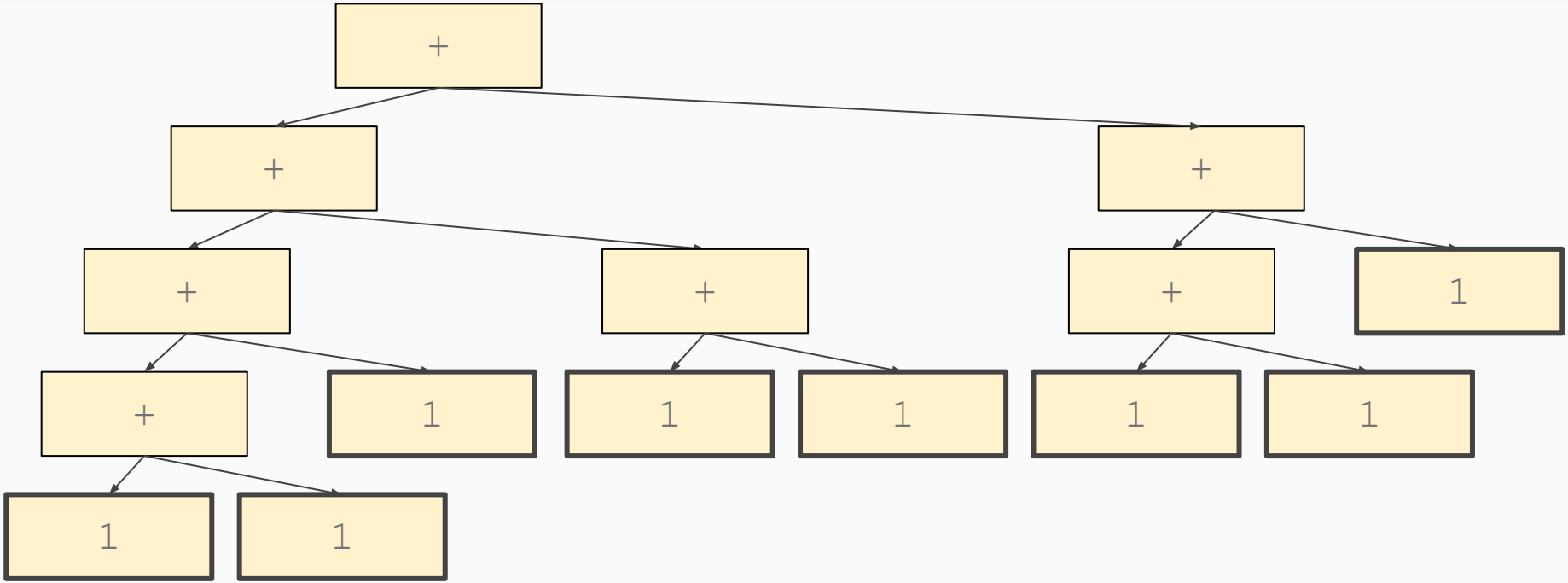
Hinweis: Benutze einen `print`-Befehl im Code.

Fibonacci-Zahlen: Rekursion



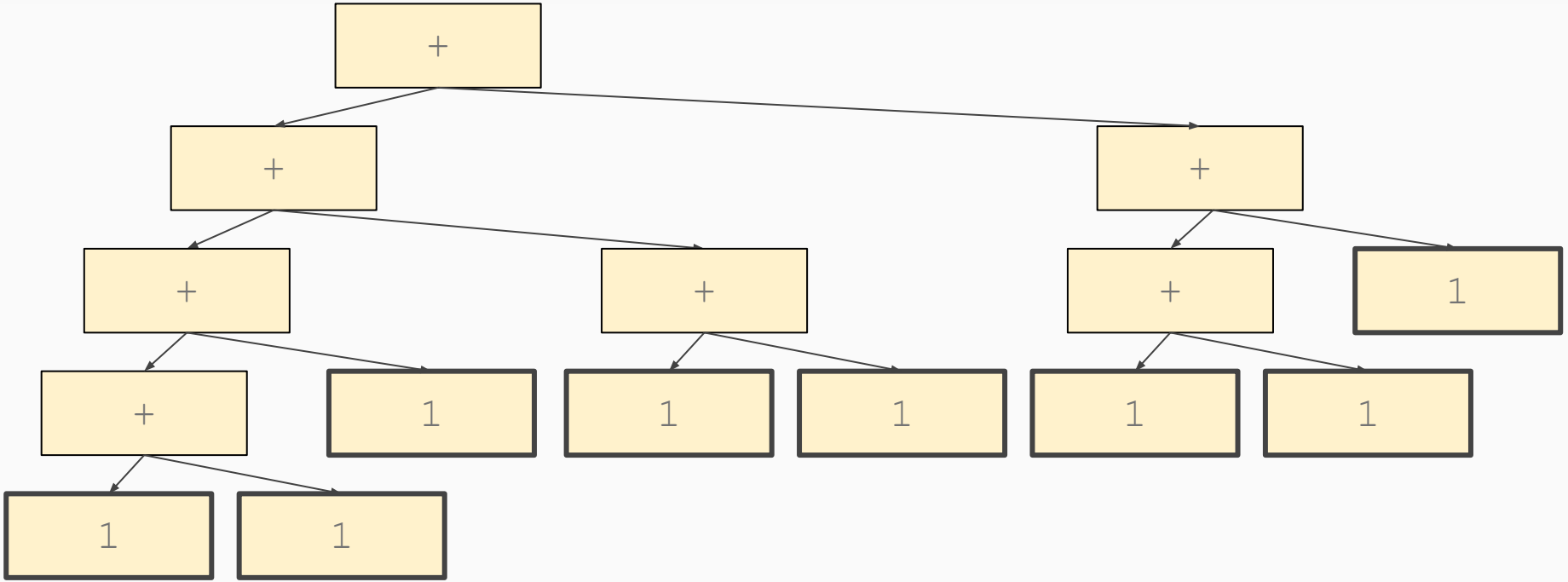
Wie viel Mal wird `fibo()` aufgerufen?

Fibonacci-Zahlen: Rekursion



Wie viel Mal wird `fibo()` aufgerufen?

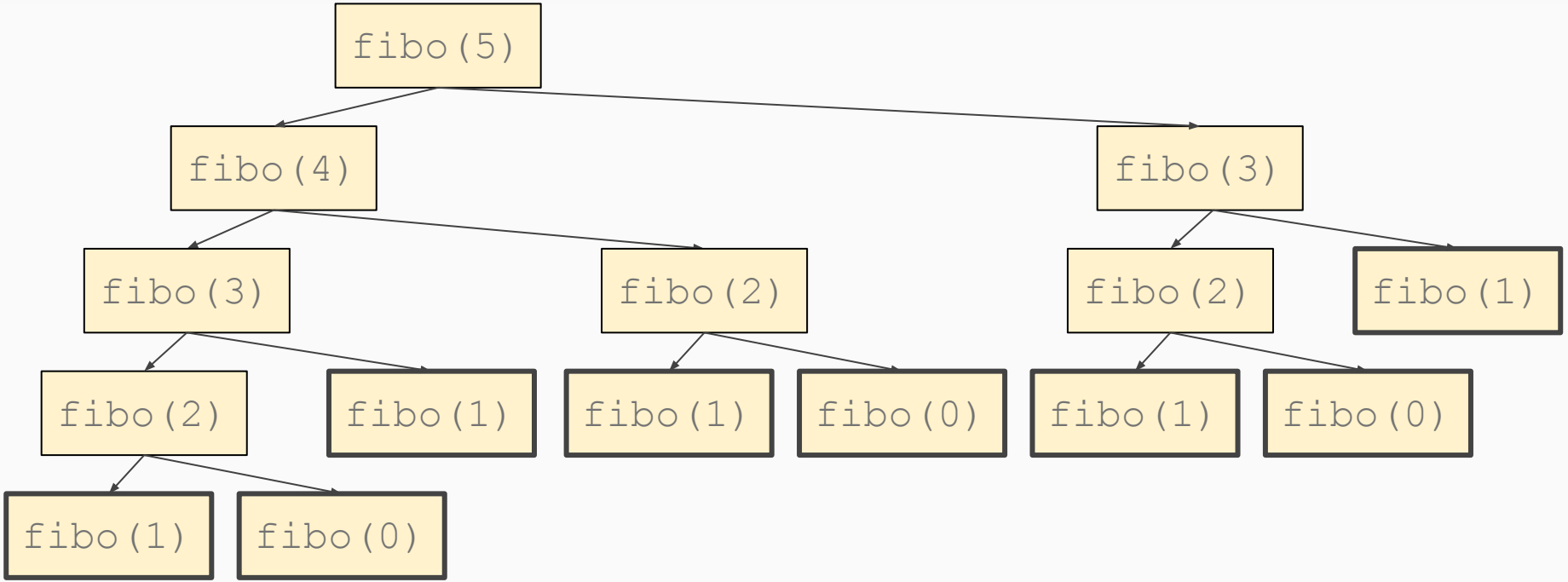
Fibonacci-Zahlen: Rekursion



Wie viel Mal wird `fibo()` aufgerufen?

`2*fibo(n)-1` Mal

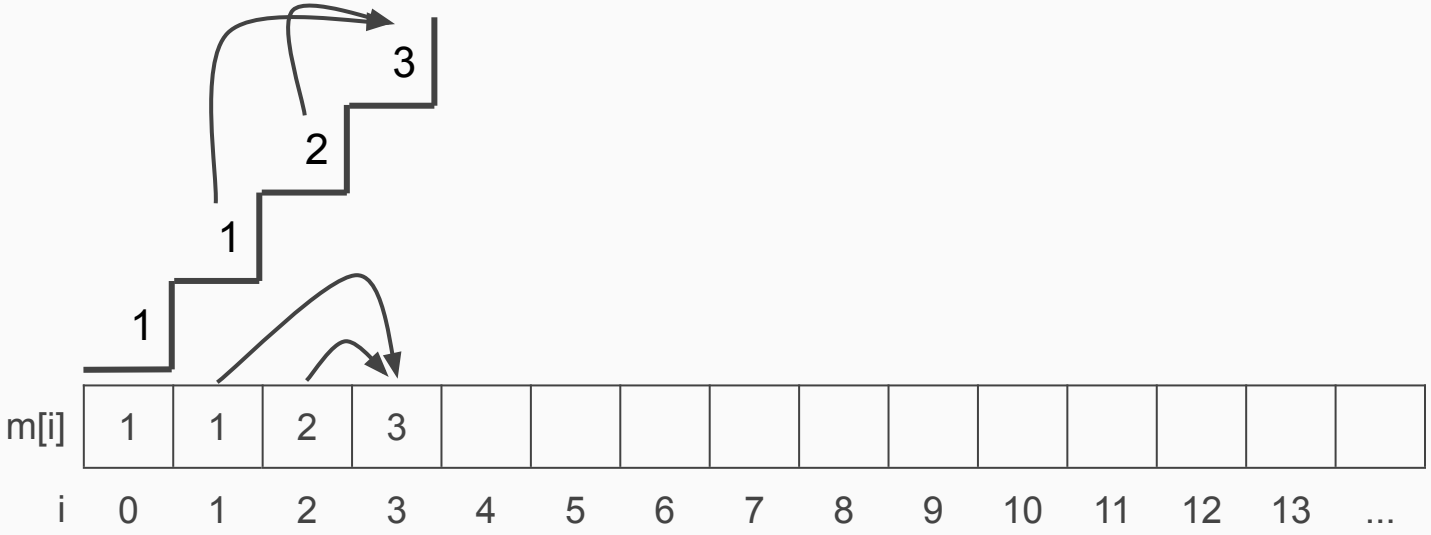
Fibonacci-Zahlen: Rekursion



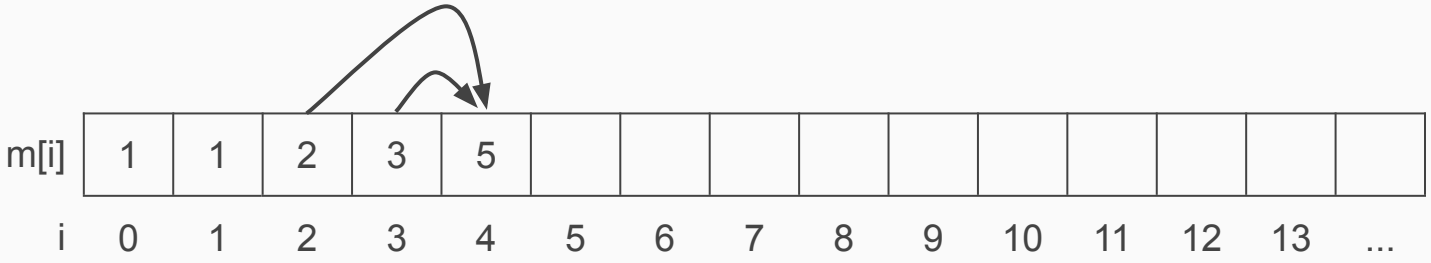
Wie viel Mal wird `fibo()` aufgerufen?

`2*fibo(n)-1` Mal

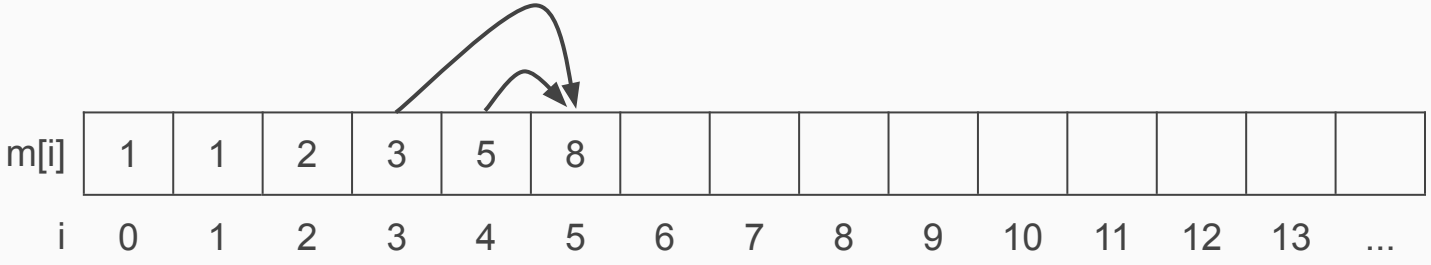
Treppensteigen: Bottom-up



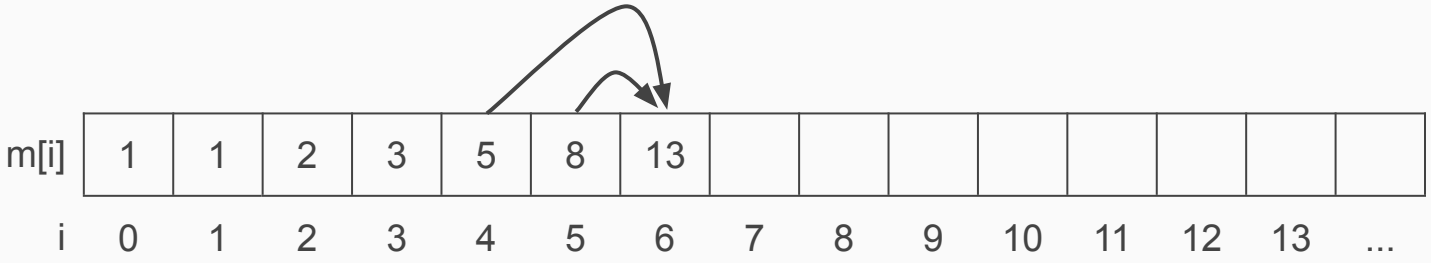
Fibonacci-Zahlen: Bottom-up



Fibonacci-Zahlen: Bottom-up

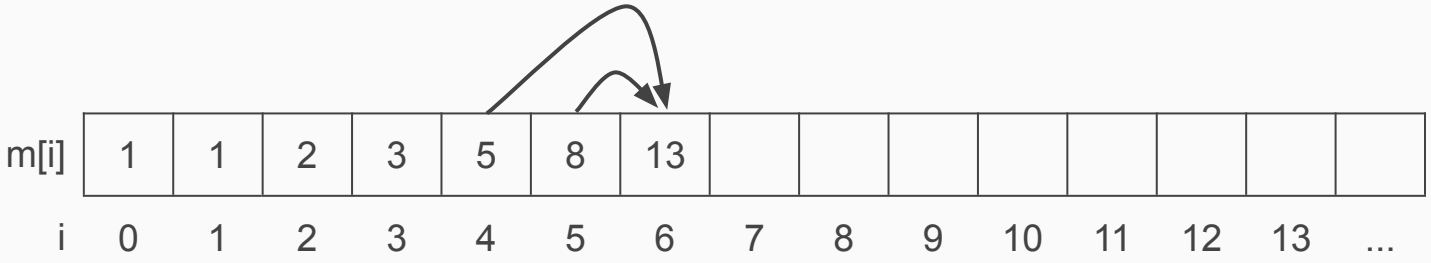


Fibonacci-Zahlen: Bottom-up



Fibonacci-Zahlen: Bottom-up

```
m[0] = 1
m[1] = 1
for i in range(2, N+1):
  m[i] = m[i-1] + m[i-2]
```



Fibonacci-Zahlen: Bottom-up

```
m[0] = 1
m[1] = 1
for i in range(2, N+1):
  m[i] = m[i-1] + m[i-2]
```

m[i]	1	1	2	3	5	8	13	21	34	55	89	144	233	377	...
i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	...

Fibonacci-Zahlen: Bottom-up

Fibonacci-Zahlen: $F(0) = 1, \quad F(1) = 1, \quad F(k) = F(k - 1) + F(k + 2)$

```
m[0] = 1
m[1] = 1
for i in range(2, N+1):
  m[i] = m[i-1] + m[i-2]
```

m[i]	1	1	2	3	5	8	13	21	34	55	89	144	233	377	...
i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	...

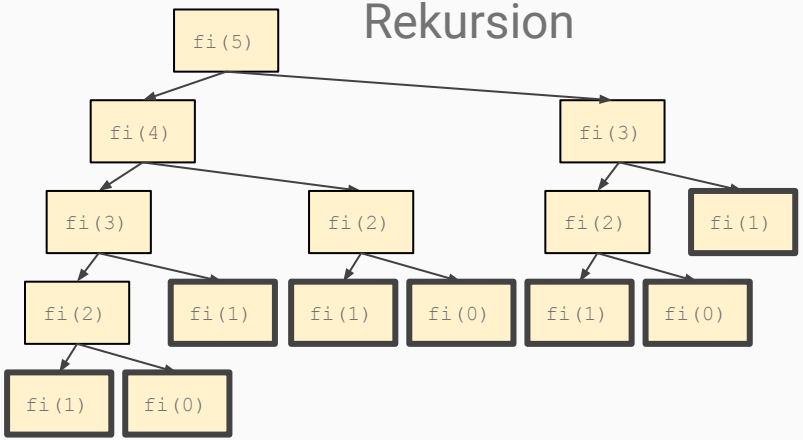
Fibonacci-Zahlen: Rekursion

Fibonacci-Zahlen: $F(0) = 1, \quad F(1) = 1, \quad F(k) = F(k - 1) + F(k + 2)$

```
def fibo(k):
  if (k == 0) or (k == 1):
    return 1
  else:
    return fibo(k-1) + fibo(k-2)
```

Fibonacci-Zahlen: Rekursion vs. Bottom-up

Rekursion



Laufzeit ~ fibo(n)

n = 10 ~ 89

n = 20 ~ 10946

Bottom-up

Laufzeit ~ n

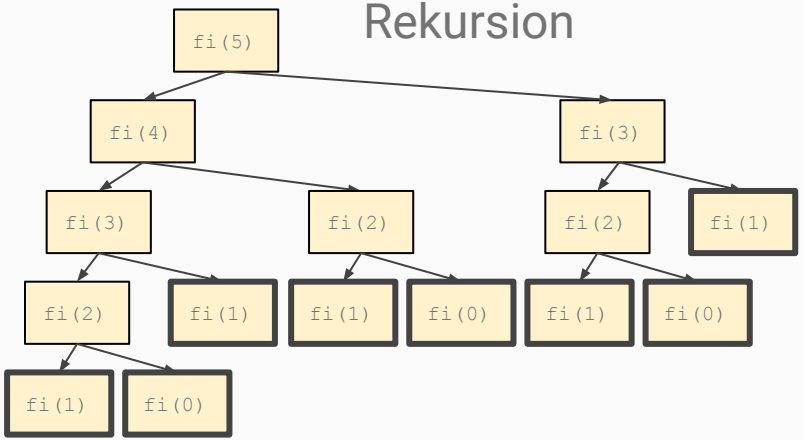
~ 10

~ 20

Für welche n können wir fibo(n) noch berechnen?

Fibonacci-Zahlen: Rekursion vs. Bottom-up

Rekursion

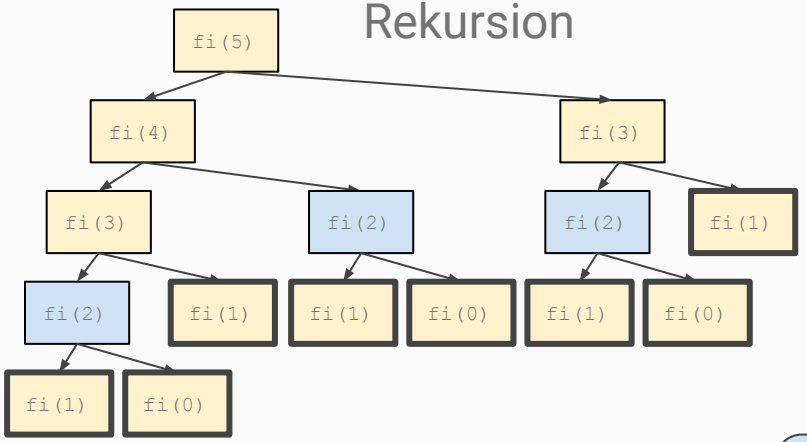


Bottom-up

	Laufzeit ~ fibo(n)	Laufzeit ~ n
n = 10	~ 89	~ 10
n = 20	~ 10946	~ 20
n = 100	~ 573147844013817084101	~ 100
n = 1000	~ $7 * 10^{208}$	~ 1000

Fibonacci-Zahlen: Rekursion vs. Bottom-up

Rekursion



Bottom-up

	Laufzeit ~ fibo(n)	Laufzeit ~ n
n = 10	~ 89	~ 10
n = 20	~ 10946	~ 20
n = 100	~ 573147844013817084101	~ 100
n = 1000	~ $7 * 10^{208}$	~ 1000

fibo(2) = 1+1 wird
21892299583455516902 6
Mal berechnet

Aufgabe: Sportler-Treppensteigen

Eine Treppe hat 20 Stufen. **Du kannst immer entweder 1, 2 oder 3 auf einmal nehmen.** Wie viele Möglichkeiten gibt es, die Treppe mit den 20 Stufen hochzugehen?

Beispiel: Du kannst 4 Stufen auf 7 Arten hochgehen:

1-1-1-1, 1-1-2, 1-2-1, 2-1-1, 2-2, 3-1, 1-3

Aufgabe: Skiferien-März

Jakub möchte im März viel skifahren. Er hat sich die Wettervorhersage angeschaut und weiss genau, wie viele Stunden Sonne es jeden Tag geben wird. **Er will so viele Sonnenstunden wie möglich geniessen, er kann aber nie an zwei aufeinanderfolgenden Tagen skifahren** -- das wäre zu anstrengend. Schreibe eine Funktion, die ihm die maximale Anzahl Ski-Sonnenstunden berechnet.

Beispiel: max_stunden([2,4,1,0,6]) = 10

(Er muss am zweiten und letzten Tag fahren.)