

Lab 4 - Kim Groen

1.1 • def ggT(x, y)

if $x > y$:

return ggT(x-y, y)

elif $x < y$:

return ggT(x, y-x)

else:

return x

• ggT(15, 10):

↓

ggT(5, 10)

↓

ggT(5, 5)

↓

5

1.2 • def fib(x)

if $x == 0$:

return 0

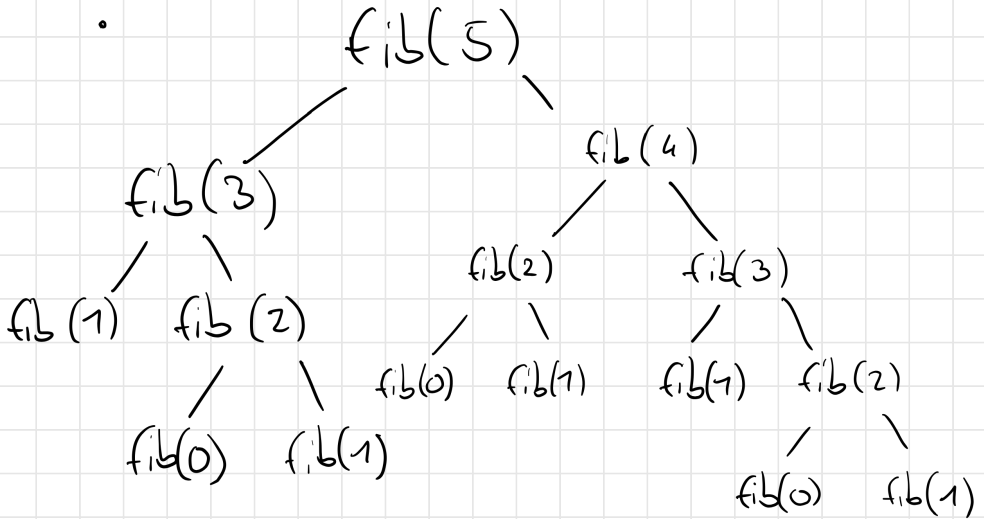
elif $x == 1$

return 1

else:

return fib(x-1) + fib(x-2)

2.



- $\text{fib}(4)$ benötigt 9 Aufrufe
 $\text{fib}(5)$ benötigt 15 Aufrufe

- exponentiell?

$$\text{fib}(6) \hat{=} 25$$

$$\text{fib}(7) \hat{=} 41$$

- Dieses, weil es nicht rekursiv ist.

2.1 funktion $\text{search}(L, n)$

| Suche True:

		if $istLeer(L) = True:$
1	1	return False
		if $nächsteZahl(L) = n:$
	1	return True

Schleppfehler: $len(L)$

besten: 1

Durchschnitt: $len(L) : 2$

2.2 $len(L) = 10$

$mitte = 10$

$wert[10] = 47 > 32$

$\hookrightarrow bs_r(L, 32, 1, 9)$

$mitte = 5$

$wert[5] = 31 < 32$

$\hookrightarrow bs_r(L, 32, 6, 9)$

$mitte = 7$

$wert[7] = 36 > 32$

$\hookrightarrow bs_r(L, 32, 6, 6)$

mitte = 6

werte[6] = 32 = 32

↳ return 6

linear: 6 Schritte

binär 4 Schritte

binär oft besser als linear