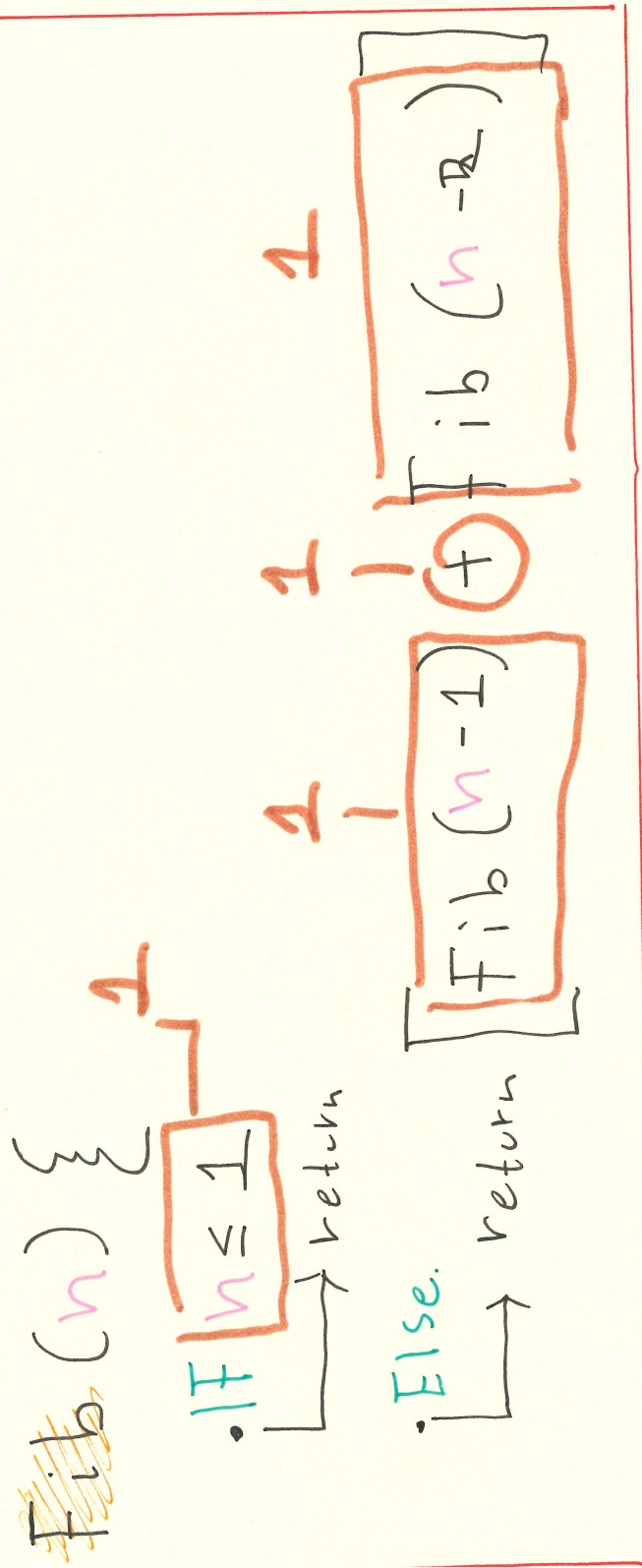


Time complexity

Recursive



Two calls to previous + C

$$\begin{aligned}
 T(n) &= 2T(n-2) + C \\
 &= 2\{2T(n-4) + C\} + C \\
 &= 4T(n-4) + 2C + C \\
 &= 4T(n-4) + 3C \\
 &= 4(2T(n-6) + 3C) + 3C \\
 &= 8T(n-6) + 4C + 3C \\
 &= 8T(n-6) + 7C
 \end{aligned}$$

$$\boxed{T(n) = 2^n}$$

$O_n = 2^n$

(Master Theorem)

$$T(n) = 2^k (T(n-2k)) + (2^k - 1) \cdot C$$

$k = \frac{n}{2}$

At $n-k=0 \rightarrow k=n$

$$T(n) = 2^n (T(0)) + (2^n - 1)C$$

$$T(n) = T(n-1) + T(n-2) + 4$$

$$\begin{aligned}
 T(0) &= T(1) = 1 \\
 T(n-2) &= T(n-1) = x
 \end{aligned}$$

$$\begin{aligned}
 T(n) &= 2x + C \\
 &= 2(T(n-2)) + C \\
 &= (T(n-2) + T(n-1)) + C \\
 &= 2x + C \\
 &= 4T(n-2)
 \end{aligned}$$