Algoritmo (per numeri di fibonacci)

$$F_n = \begin{cases} 1 & n = 1, 2 \\ F_{n-1} + F_{n-2} & n \geq 3 \end{cases}$$

formula di Binet

sezione aurea

$$x^2 = x + 1 \longrightarrow x^2 - x - 1 = 0 \quad \begin{cases} \Phi = \frac{1+\sqrt{5}}{2} \simeq 1.6\ldots \\ \hat{\Phi} = \frac{1-\sqrt{5}}{2} \simeq -0.6\ldots \end{cases}$$

complemento della sezione aurea

$$\forall n \geq 1 : F_n = \frac{1}{\sqrt{5}} \left(\Phi^n - \hat{\Phi}^n \right)$$

Dimostrazione tramite passo induttivo

$$n = 1 \implies F_n = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} - \frac{1-\sqrt{5}}{2} \right) = \frac{1}{\sqrt{5}} \left(\frac{2\sqrt{5}}{2} \right) = 1 \checkmark$$

$$n = 2 \implies F_n = \frac{1}{\sqrt{5}} \left(\left(\frac{1+\sqrt{5}}{2} \right)^2 - \left(\frac{1-\sqrt{5}}{2} \right)^2 \right)$$

$$\hookrightarrow \frac{1}{\sqrt{5}} \left(\frac{1+2\sqrt{5}+5}{4} - \frac{1-2\sqrt{5}+5}{4} \right)$$

$$\hookrightarrow \frac{1}{\sqrt{5}} \left(\frac{4\sqrt{5}}{4} \right) = 1 \ V$$

per $n \geq 3$ l'ipotesi dice che la proprietà vale fino ad $n-1$ quindi se:

$$F_n = 1/\sqrt{5}\left(\Phi^n - \hat{\Phi}^n\right) \quad e \quad F_n = F_{n-1} + F_{n-2}$$

per ipotesi induttiva allora

$$F_n = 1/\sqrt{5}\left(\Phi^{n-1} - \hat{\Phi}^{n-1}\right) + 1/\sqrt{5}\left(\Phi^{n-2} - \hat{\Phi}^{n-2}\right)$$

$$= 1/\sqrt{5}\left[\left(\Phi^{n-1} + \Phi^{n-2}\right) - \left(\hat{\Phi}^{n-1} + \hat{\Phi}^{n-2}\right)\right]$$

....

$$1/\sqrt{5}\left(\Phi^n - \hat{\Phi}^n\right) \;?$$

come ci arrivo? dimostro se é possibile che

$$\begin{cases} \Phi^n = \Phi^{n-1} + \Phi^{n-2} \\ \hat{\Phi}^n = \hat{\Phi}^{n-1} + \hat{\Phi}^{n-2} \end{cases}$$

se sono entrambe vere la dimostrazione é finita

divido per Φ^{n-2} e $\hat{\Phi}^{n-2}$

$$\begin{cases} \Phi^2 = \Phi + 1 \\ \hat{\Phi}^2 = \hat{\Phi} + 1 \end{cases}$$

$\to$ per definizione di Φ, avendo $x^2 = x+1$ la dimostrazione é verificata pseudocodice:

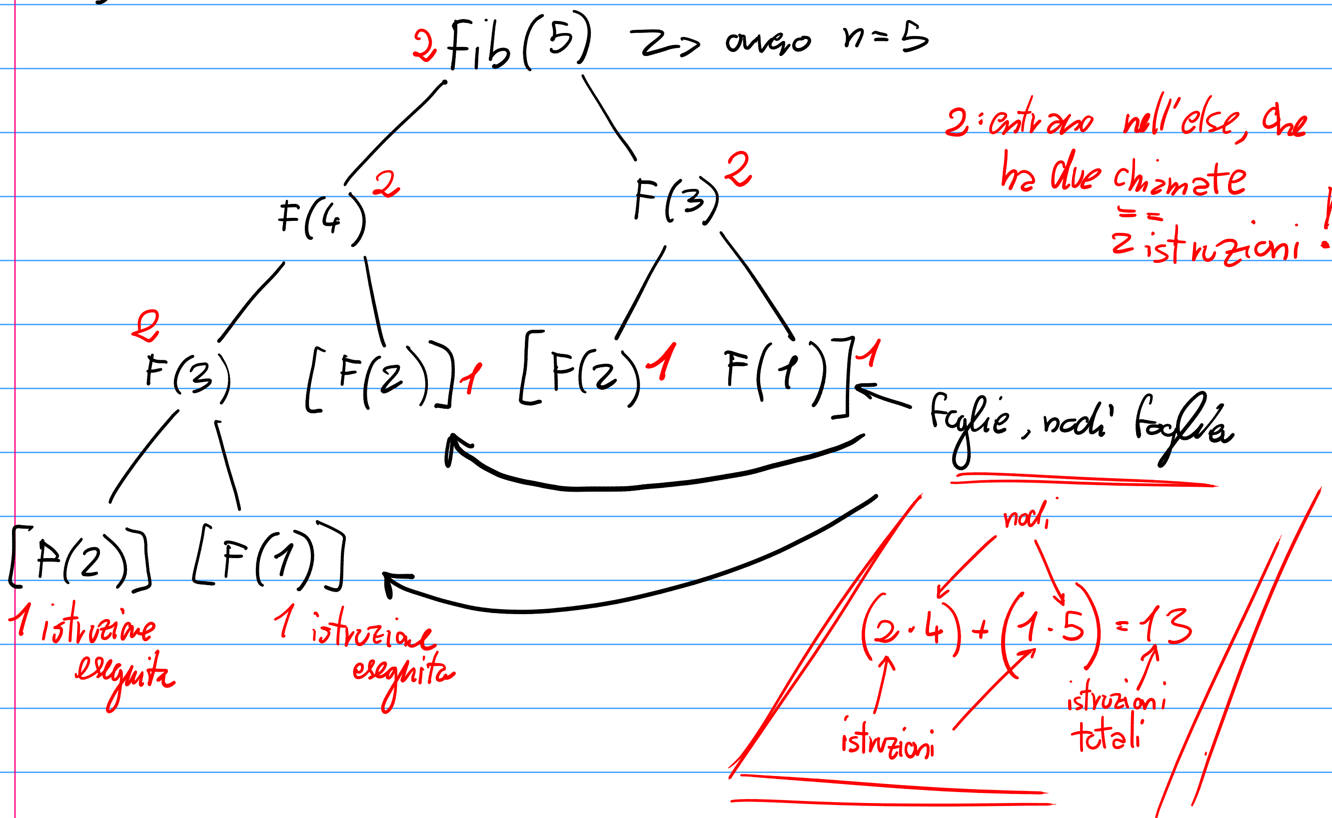
```
Fib(int n) -> int
    if n ≤ 2 then return 1;
    else return Fib(n-1) + Fib(n-2)
```

Complessità? Quante istruzioni sono eseguite?

n	$T(n)$
1	1
2	1
3	4
4	$2+4+1=7$

Albero delle ricorsioni (esempio pratico)

$n=5$



L'albero ci permette di calcolare qualunque complessità

$$T(5) = 13 \rightarrow 2 \cdot \underbrace{i(T_n)}_{\text{non foglie}} + \underbrace{f(T_n)}_{f = \text{foglie}}$$

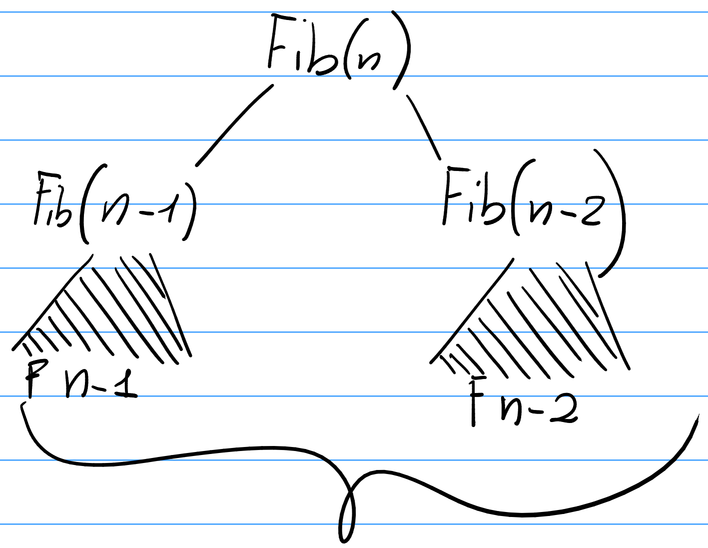
Proprietà 1 Sia T_n l'albero delle ricorsioni relativo alla chiamata Fib(n).

Allora il numero di foglie di T_n è pari a F_n (ennesimo numero di Fibonacci)

$$f(T_n) = F_n$$

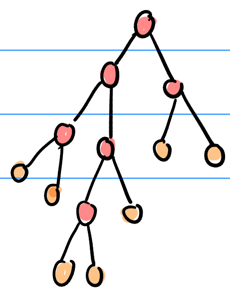
Dim: induttiva su n (meglio con il disegno

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



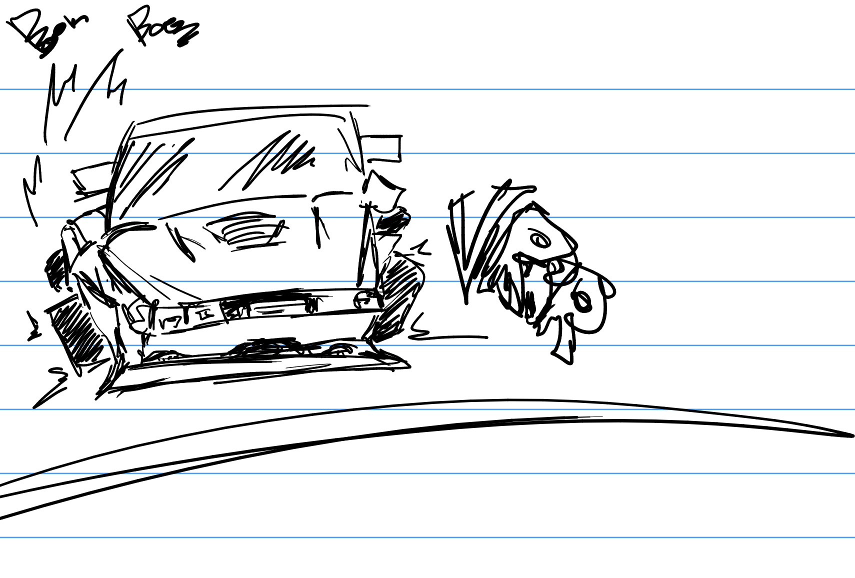
F_n = iesimo numero di Fibonacci

Proprietà 2 se ogni nodo ha esattamente due figli, allora $i(T) = f(T) - 1$



● $= i(T_n) = f(T_n) - 1$ (albero binario)

● $= F(n)$ (caso fibonacci)



Fib(int n) → int
if $n \leq 2$ then return 1;
else return Fib(n-1) + Fib(n-2)

ricordo $T(n)$ = numero di istruzioni mandate in esecuzione!

$$\begin{cases} T(n) = 2 + T(n-1) + T(n-2) & \text{quando } n \geq 3 \\ 1 & \text{quando } n = 1,2 \end{cases}$$

ricorsiva come l'algoritmo 😑... prof wtf

formula generale

foglie

$$\boxed{T(n) = 2 \cdot i(T_n) + f(T_n)}$$

nodi non foglie

a) $f(T_n) = F_n$

b) $i(T_n) = f(T_n) - 1$

riprendo pagine successive

quindi posso semplificare la formula generale con:

$$T(n) = 2(F_n - 1) + F_n = 3F_n - 2$$

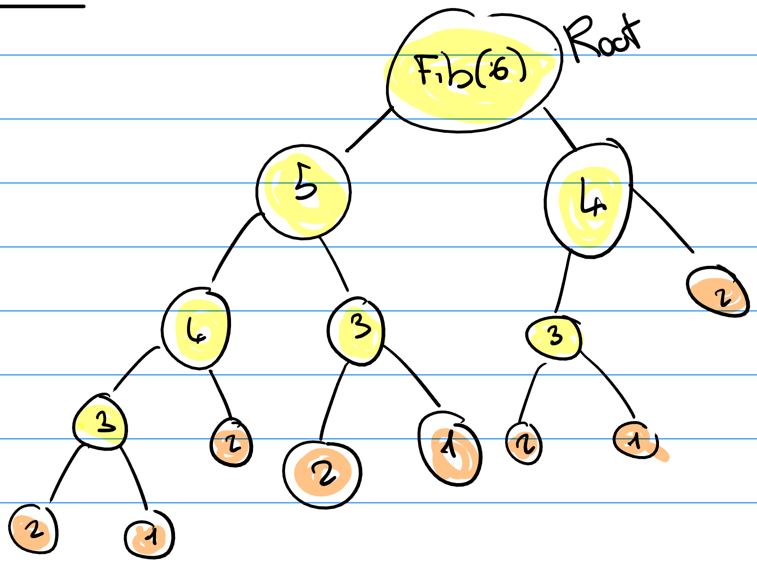
complessità cresce "come i numeri di Fibonacci"

$T(n) \simeq F_n \qquad \forall n \geq 6$

$F_n \geq 2^{n/2}$

Ind. su n (verifichiamo)

<u>Base</u> : $n = 6, 7$



$n \geq 8$

$$F_n = \underbrace{F_{n-1}}_{2^{(n-1)/2}} + \underbrace{F_{n-2}}_{2^{(n-2)/2}}$$

$$\geq_{I} 2^{(n-1)/2} + 2^{(n-2)/2}$$

$$2^{\frac{n}{2}}\left(\frac{1}{\sqrt{2}} + \frac{1}{2}\right) \geq 2^{n/2}$$

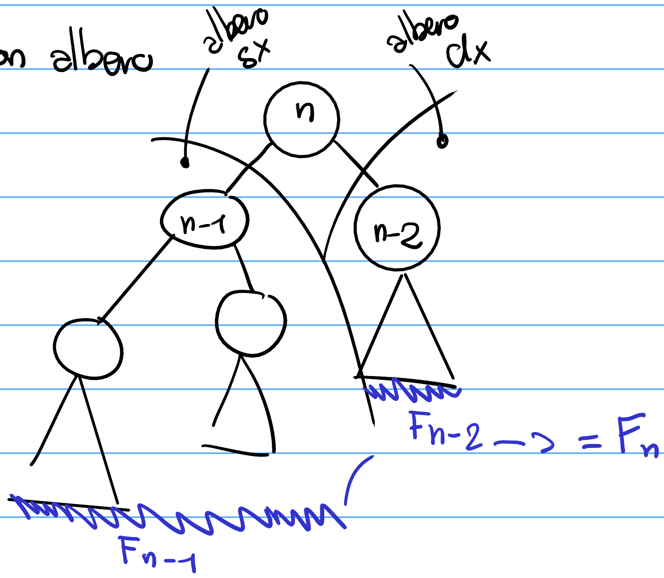
riprendo formule

$f(T_n) = F_n$? numero figli $= F_n$?

<u>base</u> $n = 1, 2$ —> non ci sono nodi foglia ldl

$n \geq 3$

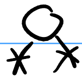
L> ipotesi con albero



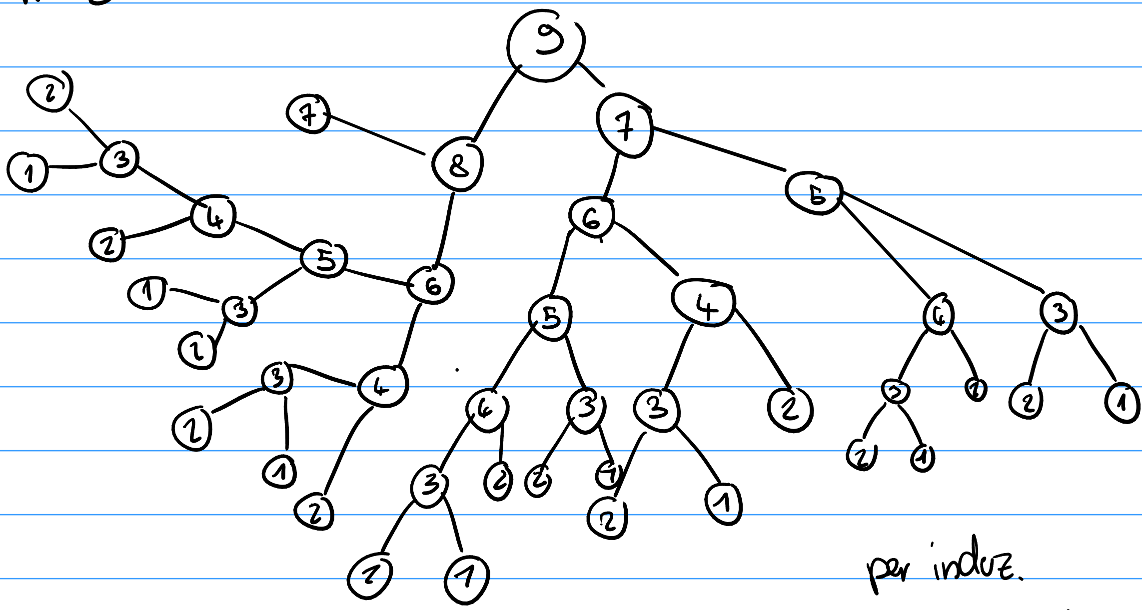
totale nodi foglia = somma numero nodi foglia sx e dx

Se T é un albero binario dove ogni vertice interno ha esattamente 2 figli:
allora $i(T) = f(T) - 1$

Dimostrazione (ind sulla grandezza dell'albero)

<u>Base</u> $n = 1, 2$  non ha nodi figli!

$n = 9$



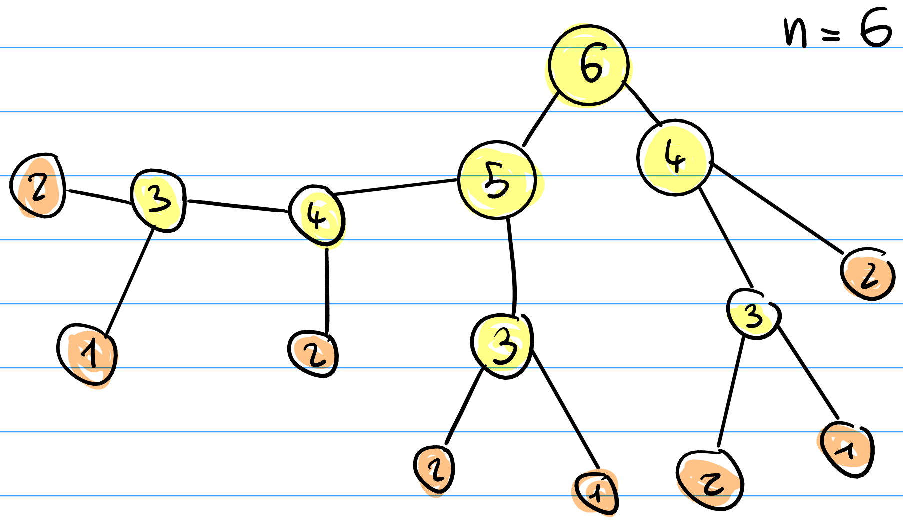
per induz.
$n = n$ nodi

$$i(T) = f(T) - 1$$

$$i(T) = \begin{cases} i(T) = i(T') + 1 \\ f(T) = f(T') + 1 \end{cases}$$

$$i(T) = \begin{cases} i(T) = i(T') + 1 = f(T') \\ f(T) = f(T') + 1 \end{cases}$$

la domanda del secolo per algoritmi complessi: l'algoritmo finirà?



Fib3(int n) -> int

1. alloca spazio per un array F di n interi
2. F[1] = F[2] = 1;
3. for i = 3 to n
4. F[i] = F[i-1] + F[i-2];
5. return F[n];

n = 3 + (n-2) + (n-1)

4 3

1,2,5

= 2·n -> 2 è non considerabile => n