· FORMULA BINET

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

→ SEZIONE AUREA $\phi = \frac{1+\sqrt{5}}{2} \approx 1,618...$ * $\hat{\phi} = \frac{1-\sqrt{5}}{2} \approx -0.618... = 1 - \phi$

x □ area $x+1$ ⇝ $x^2 = x+1$ → $x^2 - x - 1 = 0$ — $x_1 \approx 1,618...$, $x_2 \approx -0,618...$ *

• $F_{n \geq 1}$, $n \in \mathbb{N}$: $\boxed{F_n = \frac{1}{\sqrt{5}}(\phi^n - \hat{\phi}^n)}$ *formula di Binet* !

· FIB_1 (int n) → int
1. RETURN $\frac{1}{\sqrt{5}}(\phi^n - \hat{\phi}^n)$

$T(n) = 1$; $S(n) = 1$ complessità SPAZIALE !

NON è CORRETTO !! ↪ FIB_1(18) = 2583 NO !! problemi d'ottimizzazione

---

· dim. formula di Binet

CASO BASE

· $n = 1$
$F_1 = 1$? → $\frac{1}{\sqrt{5}}(\phi^1 - \hat{\phi}^1) = \left(\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2} - \frac{1-\sqrt{5}}{2}\right)^2\right) = \frac{1}{\sqrt{5}}\left(2\frac{\sqrt{5}}{2}\right) = 1$ ✓

· $n = 2$
$F_2 = 1$? ⇝ $\frac{1}{\sqrt{5}}[\phi^2 - \hat{\phi}^2] = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^2 - \left(\frac{1-\sqrt{5}}{2}\right)^2\right] = ... = 1$ ✓

PASSO INDUTTIVO

· $n \geq 3$ assumo che $F_n = \frac{1}{\sqrt{5}}(\phi^n - \hat{\phi}^n)$ valga fino a $n-1$ e dimostro per n

[per definizione] $F_n = F_{n-1} + F_{n-2}$

[per ip. induttiva] $\doteq \frac{1}{\sqrt{5}}(\phi^{n-1} - \hat{\phi}^{n-1}) + \frac{1}{\sqrt{5}}(\phi^{n-2} - \hat{\phi}^{n-2}) = \frac{1}{\sqrt{5}}\left[(\phi^{n-1} - \hat{\phi}^{n-1}) + (\phi^{n-2} - \hat{\phi}^{n-2})\right] =$

$= \frac{1}{\sqrt{5}}\left[\underbrace{(\phi^{n-1} + \phi^{n-2})}_{*} - \underbrace{(\hat{\phi}^{n-1} + \hat{\phi}^{n-2})}_{*}\right]$

* [per SEZ. AUREA]
$(\phi^n = \phi^{n-1} + \phi^{n-2}$: divido per ϕ^{n-2} ⇒ $\phi^2 = \phi + 1)$; $(\hat{\phi}^n = \hat{\phi}^{n-1} + \hat{\phi}^{n-2}$: divido per $\hat{\phi}^{n-2}$: $\hat{\phi}^2 = \hat{\phi} + 1)$ } $x^2 = x + 1$

· ~~ALGOR~~ RICORSIVO

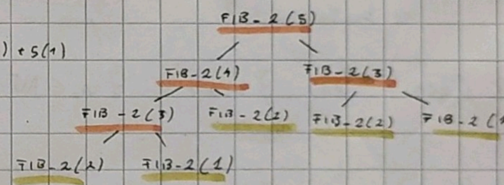
· FIB_2 (int n) → int
1. IF $n \leq 2$ THEN RETURN 1
2. ELSE RETURN FIB_2(n-1) + FIB_2(n-2)

- è CORRETTO (è la definizione)
- $T(n) = 2^n$ (vedi prossima pagina !) *
- $S(n) = n$ (lineare)

· ALBERO delle RICORSIONI

$n = 5$

$T(n) = 4(2) + 5(1)$



n	T(n)
1	1 caso base RIC
2	1
3	1+1+2=4
4	2+4+1=7
⋮	

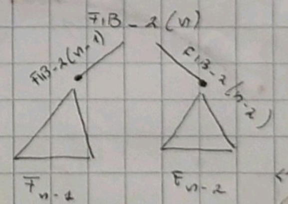
⇝ $T(n) = \underset{\text{nodi interni}}{2\,i(T_n)} + \underset{\text{nodi foglie}}{f(T_n)}$

• i. $\boxed{f(T_n) = F_n}$ T_n albero delle ricorsioni

num. foglie — num. fibonacci

- dim !!

CASO BASE
· $n = 1$ $f(T_1) = F_1 = 1$ ○ root
· $n = 2$ $f(T_2) = F_2 = 1$ ○ root



PASSO INDUTTIVO
· $n \geq 3$ $f(T_n) = f(T_{n-1}) + f(T_{n-2}) \overset{\text{[per ip. induttiva]}}{=} F_{n-1} + F_{n-2} = F_n$

• ii. $\boxed{i(T) = f(T) - 1}$ T albero binario !

nodi interni — foglie

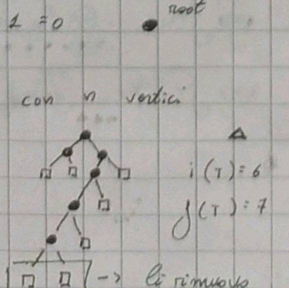
♥ [ip. induttiva] ① $i(T') = f(T') - 1$
▲ [per costruzione] ② $i(T) = i(T') + 1$
▲ [per costruzione] ③ $f(T) = f(T') + 1$

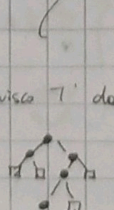
- dim !! [induzione sul numero di vertici di T]

CASO BASE
· $n = 1$ $0 = 1 - 1 = 0$ ● root

quindi $i(T) =$ [per ②] $= i(T') + 1 =$
[per ①] $= f(T') - 1 + 1 =$
[per ③] $= f(T) - 1$ ☺

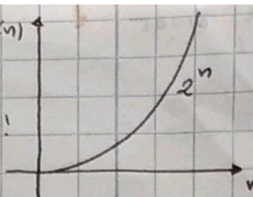
PASSO INDUTTIVO
· $n \geq 3$ T con n vertici — costruisco T' da T





allora ho $n-2$ vertici e vale l'IP induttiva $i(T') = f(T') - 1$ ♥

* $T(n) = 2 + T(n-1) + T(n-2)$
  $T(n) = 2 i(T_n) + f(T_n) = 2(F_n - 1) + F_n = 3F_n - 2 \rightsquigarrow \boxed{T(n) \approx F_n = 2^n}$
  CRESCE ESPONENZIALMENTE! è una cosa brutta ¨



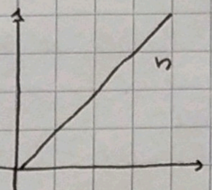
dim ¨

PASSO INDUTTIVO $F_n \geq 2^{\frac{n}{2}}$ per $n \geq 6$!
- $F_n = F_{n-1} + F_{n-2}$ [def. FIBONACCI]
- $F_n \geq 2^{\frac{n-1}{2}} + 2^{\frac{n-2}{2}}$ [per ip. induttiva] $\geq 2^{\frac{n}{2}} \cdot 2^{-\frac{1}{2}} + 2^{\frac{n}{2}} \cdot 2^{-1} \rightsquigarrow 2^{\frac{n}{2}} \underbrace{\left(2^{-\frac{1}{2}} + 2^{-1}\right)}_{\geq 1} \Rightarrow F_n \geq 2^{\frac{n}{2}}$ ¨

· ITERATIVO

- FIB_3 (int n) -> int
  1. alloca spazio per un array n-dimensionale (n interi)
  2. F[1] <- F[2] <- 1
  3. FOR i=3 TO n $(n-3)+2 = n-1$
  4. F[i] <- F[i-1] + F[i-2] $(n-3)+1 = n-2$
  5. RETURN F[n]

- FIB_4 (int n) -> int
  1. a <- 1 ; b <- 1
  2. FOR i=3 TO n
  3. c <- a+b
  4. a <- b
  5. b <- c
  6. return b



- FIB_3 : $S(n) = n$ (array)
- FIB_4 : $S(n) = c$ (costante, 3 VARIABILI!)
- $n \geq 3 \rightsquigarrow T(n) = 3 + (n-1) + (n-2) = 2n \Rightarrow \boxed{T(n) \approx n}$

---

for i=1 TO n ← n+1 volte (+1 ultimo controllo)
  // BLOCCO INTERNO ← eseguito n volte

for i=K TO n ← $(n-K)+2$
  // BLOCCO INTERNO ← $(n-K)+1$ volte

---

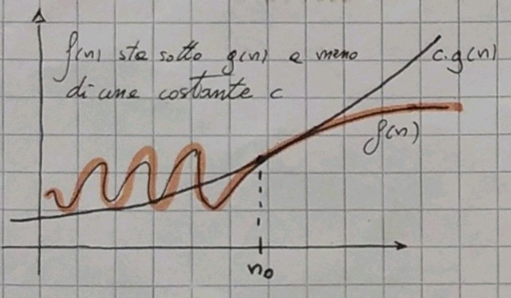
comportamento di $T(n)$ per n sufficientemente grande ⁺⁺ ASINTOTICAMENTE ⁺⁺

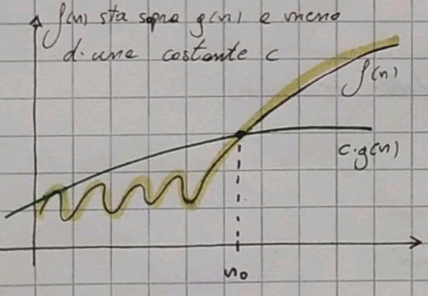
i. $f(n) = O(g(n)) \iff \exists c > 0,\ \underbrace{\exists n_0 \in \mathbb{N} \ni' \forall n \geq n_0}_{\text{per n sufficientemente grande}} : f(n) \leq c \cdot g(n)$
ne basta uno!

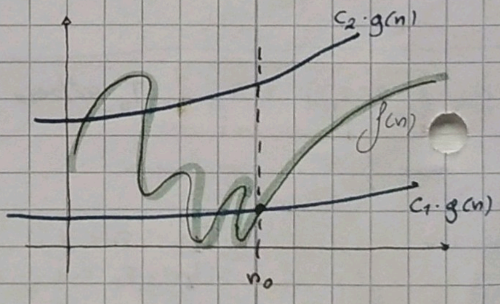
ii. $f(n) = \Omega(g(n)) \iff \exists c > 0,\ \exists n_0 \in \mathbb{N} \ni' \forall n \geq n_0 : f(n) \geq c \cdot g(n)$

iii. $f(n) = \Theta(g(n)) \iff \exists c_1 > 0,\ \exists c_2 > 0,\ \exists n_0 \in \mathbb{N} \ni' \forall n \geq n_0 : c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$

f e g si comportano in modo asintoticamente equivalente!

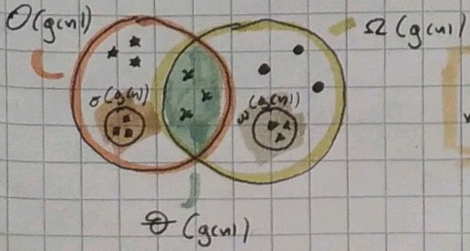






iv. $f(n) = o(g(n)) \iff \forall c > 0,\ \exists n_0 \in \mathbb{N} \ni' \forall n \geq n_0 : f(n) < c \cdot g(n)$
per ogni! condizione molto forte

v. $f(n) = \omega(g(n)) \iff \forall c > 0,\ \exists n_0 \in \mathbb{N} \ni' \forall n \geq n_0 : f(n) > c \cdot g(n)$



$$\boxed{\lim_{n \to \infty} \frac{f(n)}{g(n)}} = \begin{cases} 0 & \iff f(n) = o(g(n))\ [\Rightarrow f(n) = O(g(n))] \\ \ell\ (0 < \ell < +\infty) & \Rightarrow f(n) = \Theta(g(n)) \\ +\infty & \iff f(n) = \omega(g(n))\ [\Rightarrow f(n) = \Omega(g(n))] \end{cases}$$

- $f(n) = o(g(n)) \Rightarrow f(n) = O(g(n))$
- se $f(n) = O(g(n)) \wedge f(n) = \Omega(g(n)) \Rightarrow f(n) = \Theta(g(n))$
- $f(n) = \omega(g(n)) \Rightarrow f(n) = \Omega(g(n))$

- $\boxed{f(n) = O(g(n)) \iff g(n) = \Omega(f(n))}$

dim

$(\Rightarrow)$ $f(n) = O(g(n))$ ... $\exists c > 0,\ \exists n_0 \in \mathbb{N} \ni' \forall n \geq n_0 : f(n) \leq c \cdot g(n)$ ipotesi

$g(n) = \Omega(g(n))$ ... $\exists c' > 0,\ \exists n_0' \in \mathbb{N} \ni' \forall n \geq n_0' : g(n) \geq c' \cdot f(n)$ tesi

- $c' = \frac{1}{c}$
- $n_0' = n_0$

[per ipotesi] $g(n) \geq \frac{1}{c} \cdot f(n) \Rightarrow c' = 1/c$

PROPRIETÀ TRANSITIVA

- $\boxed{f(n) = O(g(n)) \wedge g(n) = O(h(n)) \Rightarrow f(n) = O(h(n))}$

dim

- $n_3 = \max\{n_1, n_2\}$
- $c_3 = c_1 \cdot c_2$

(ip. 1) : $\exists c_1 > 0,\ \exists n_1 \in \mathbb{N} \ni' \forall n \geq n_1 : f(n) \leq c_1 \cdot g(n)$

(ip. 2) : $\exists c_2 > 0,\ \exists n_2 \in \mathbb{N} \ni' \forall n \geq n_2 : g(n) \leq c_2 \cdot h(n)$

(tesi) : $\exists c_3 > 0,\ \exists n_3 \in \mathbb{N} \ni' \forall n \geq n_3 : f(n) \leq c_3 \cdot h(n)$

(ip. 1) vale da n_1 in poi
(ip. 2) vale da n_2 in poi $\Rightarrow$ $n_3 = \max\{n_1, n_2\}$ devono valere entrambe

quindi $\forall n \geq n_3 : f(n) \leq c_1 \cdot g(n) \wedge g(n) \leq c_2 \cdot h(n)$

$\Downarrow$

$f(n) \leq \underbrace{c_1 \cdot c_2}_{c_3!} \cdot h(n)$

- $\boxed{(i)\ o(g(n)) \cap \Omega(g(n)) = \emptyset}$
- $\boxed{(ii)\ \omega(g(n)) \cap O(g(n)) = \emptyset}$

dim (i) per ASSURDO!

$\exists f(n) : f(n) = o(g(n)) \wedge f(n) = \Omega(g(n))$

[ip. 1] $\forall c > 0,\ \exists n_0 \in \mathbb{N} \ni' \forall n \geq n_0 : f(n) < c \cdot g(n)$

[ip. 2] $\exists c' > 0,\ \exists n_0' \in \mathbb{N} \ni' \forall n \geq n_0' : f(n) \geq c' \cdot g(n)$

valgono entrambe da $n = \max\{n_1, n_2\}$
se (ip. 1) vale $\forall c > 0$ varrà anche per la c' di (ip. 2)
MA questo È ASSURDO

- $\boxed{f(n) = o(g(n)) \iff \lim_{n \to \infty} \frac{f(n)}{g(n)} = 0}$

dim

$\forall c > 0,\ \exists n_0 \in \mathbb{N} \ni' \forall n \geq n_0 : f(n) < c \cdot g(n)$

$\left| \frac{f(n)}{g(n)} \right| < c \rightsquigarrow \left| \frac{f(n)}{g(n)} - 0 \right| < c$

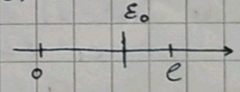
- $\boxed{\log n = O(\sqrt{n})}$

⚠ $\lim_{x \to \infty} \frac{f(x)}{g(x)} = \left[\frac{\infty}{\infty}\right] \left[\frac{0}{0}\right] = \lim_{x \to \infty} \frac{f'(x)}{g'(x)}$ ⚠

$\lim_{n \to \infty} \frac{\log n}{\sqrt{n}} = \left[\frac{\infty}{\infty}\right] = \lim_{n \to \infty} \frac{1/n}{\frac{1}{2} n^{-\frac{1}{2}}} = \lim_{n \to \infty} \frac{2\sqrt{n}}{n} = \lim_{n \to \infty} 2\sqrt{\frac{n}{n^2}} = \lim_{n \to \infty} \frac{2}{\sqrt{n}} = 0 \Rightarrow \log n = o(\sqrt{n})$
$\Rightarrow \log n = O(\sqrt{n})$

- $\boxed{\lim_{n \to \infty} \frac{f(n)}{g(n)} = \ell,\ \ \ell \in \mathbb{R} \setminus \{0\} \Rightarrow f(n) = \Theta(g(n))}$

$\varepsilon_0 \in (0, \ell)$
$\ell - \varepsilon_0 > 0$
$\ell + \varepsilon_0 > 0$



- $n_0 = n_0$
- $c_1 = \ell - \varepsilon_0$
- $c_2 = \ell + \varepsilon_0$
- $\varepsilon = \varepsilon_0$

dim.

$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \ell$ $(0 < \ell < +\infty)$, $\forall \varepsilon > 0\ \exists n_0 \in \mathbb{N} \ni' \forall n \geq n_0 : \left| \frac{f(n)}{g(n)} - \ell \right| < \varepsilon$

$\rightsquigarrow -\varepsilon < \frac{f(n)}{g(n)} - \ell < \varepsilon \rightsquigarrow \ell - \varepsilon < \frac{f(n)}{g(n)} \rightsquigarrow (\ell - \varepsilon)\, g(n) < f(n) < (\ell + \varepsilon)\, g(n)$

- $\boxed{(n+a)^b = \Theta(n^b)}$, $b > 0 \rightsquigarrow \lim_{n \to \infty} \frac{(n+a)^b}{n^b} = \lim_{n \to \infty} \left[\frac{n+a}{n}\right]^b = \left[\frac{\infty}{\infty}\right] \lim_{n \to \infty} \left(1 + \cancel{\frac{a}{n}}^{\,0}\right)^b = 1 \in \mathbb{R}$

- $\boxed{\sqrt{n+10} = \Theta(\sqrt{n})}$

$(n+10)^{1/2}$

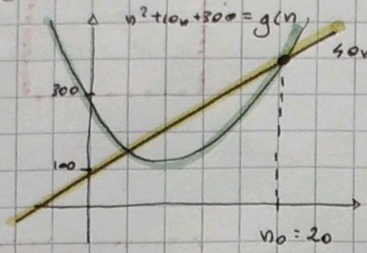
---

il logaritmo ~~schi~~ schiaccia la funzione verso il basso ⚠

$1 \quad \log\log n \quad \log n \quad \sqrt{n} \quad n \quad n\log n \quad n\sqrt{n} \quad n^2 \quad n^2\log n \quad n^2\sqrt{n} \quad n^3 \quad b^n \quad n! \quad n^n$

⚠ sono il male assoluto ⚠

$\underbrace{40n + 100}_{f(n)} = O(\underbrace{n^2 + 10n + 300}_{g(n)})$ ~> $\exists c>0 \ \exists n_0 \in \mathbb{N} \ \ni' \forall n \geq n_0 : 30n + 100 \leq c(n^2 + 10 + 300)$



- fisso $c = 1$ ~> $30n + 100 \leq (n^2 + 10 + 300)$ [provo a caso fissando $c = 1$]
  cerco l'intersezione
  $30n + 100 = n^2 + 10n + 300$
  $n^2 - 30n + 200 = 0$
  $\begin{cases} n_1 = 10 \\ n_2 = 20 \end{cases} \Rightarrow n_0 = 20$: da n_0 in poi vale che $f(n) \leq c \cdot g(n)$ $\forall n \geq 20$ e $c = 1$

$\frac{1}{2}n^2 - 3n = O(n^2)$ ~> $\exists c > 0, \exists n_0 \in \mathbb{N} \ \ni' \forall n \geq n_0 : \frac{1}{2}n^2 - 3n \leq c(n^2)$

inizio da ciò che voglio dimostrare!!!

n è sempre ≥ 0 ! impongo che sia ≤ 0 così avrei un risultato so che è ≤ 3

$\frac{1}{2}n^2 - 3n \leq c(n^2)$ ~> $\frac{1}{2}n - 3 \leq cn$ ~> $\frac{1}{2}n - cn \leq 3 \iff n\left(\frac{1}{2} - c\right) \leq 3$ ↳ $\left(\frac{1}{2} - c\right) \leq 0$ — $\boxed{c = \frac{1}{2}, \ n_0 = 1}$

$\frac{1}{2}n^2 - 3n = \Omega(n^2)$ ~> $\exists c > 0, \exists n_0 \in \mathbb{N} \ \ni' \forall n \geq n_0 : \frac{1}{2}n^2 - 3n \geq c \cdot n^2$

$\cdot \frac{1}{2}n - 3 \geq cn \iff \frac{1}{2}n - cn \geq 3 \iff n\underbrace{\left(\frac{1}{2} - c\right)}_{>0} \geq 3$ → $\left(\frac{1}{2} - c\right) > 0 \iff c < \frac{1}{2} \wedge c > 0$ (def!)

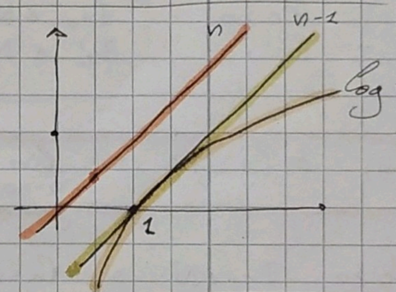
- $n \geq \frac{3}{\frac{1}{2} - c} = \frac{6}{1 - 2c}$ ~> $n \geq \frac{6}{1-2c}$

- impongo $c = \frac{1}{4}$ e ottengo $n_0 = 7$ $\boxed{c = \frac{1}{14}, \ n_0 = 7}$

$\boxed{\log n = O(n)}$ $\boxed{\log(n) = O(\sqrt{n})}$ $\boxed{\log n = o(\sqrt{n})}$

$\forall n \geq 1 : \log n \leq n - 1 \leq n \Rightarrow \log n \leq n$ [$c = 1$ e $n_0 = 1$]

$\boxed{n \log n = O(n^2)}$ $\boxed{n^k \log n = O(n^{k+1})}$



$\sqrt{n + 10} = \Theta(\sqrt{n})$ $\exists c_1 > 0, \exists c_2 > 0, \exists n_0 \in \mathbb{N} \ \ni' \forall n \geq n_0 : c_1\sqrt{n} \leq \sqrt{n+10} \leq c_2\sqrt{n}$

~> $c_1^2 n \overset{①}{\leq} n + 10 \overset{②}{\leq} c_2^2 n$

$\boxed{\begin{array}{l} \cdot \ c_2 = \sqrt{2} \\ \cdot \ c_1 = 1 \\ \cdot \ n_0 = 10 \end{array}}$ scelgo il più grande!

① $(c_1^2 - 1)n \leq 10$
impongo che $(c_1^2 - 1) \leq 0$
$c_1^2 \leq 1$
$c_1 \leq 1 \Rightarrow \begin{cases} c_1 = 1 \\ n_0 = 1 \end{cases}$

② $(n + 10) \leq c_2^2 n$
$n(c_2^2 - 1) \geq 10 \iff n \geq \frac{10}{c_2^2 - 1} \iff n \geq \frac{10}{\sqrt{2}^2 - 1}$
impongo $(c_2^2 - 1) > 0$
$c_2 > 1$ ↓ $c_2 = \sqrt{2}$

$\boxed{\sqrt{n} = \Theta(n)}$

$n\sqrt{n} = O(9^{\log_3 n})$ ~> $9^{\log_3 n} = (3^2)^{\log_3 n} = 3^{2\log_3 n^2} = n^2$!!

$\Downarrow$

$\boxed{n\sqrt{n} = O(n^2)}$ boh è ovvio ~ [$n \leq n^2$; $\sqrt{n} \leq n$; $n\sqrt{n} \leq n^2$]

$\boxed{n! = O(n^n)}$ assumo $c = 1$ $n! = \underbrace{1 \cdot 2 \cdot 3 \cdot \_ \cdot n}_{\text{sono tutti} \leq n} = \prod_{i=1}^{n} i \leq n^n = n \cdot n \cdot \_\_ \cdot n$
$\Rightarrow n! \leq (n^n)$

$\boxed{n! = \Omega(2^n)}$ $n! = 1 \cdot 2 \cdot 3 \cdot \_ \cdot n$
$\geq 2 \ \ldots \ \geq 2$
$= 1 \cdot \underbrace{2 \cdot \_ \cdot 2}_{(n-1) \text{ volte}}$ → $n! \geq 2^{n-1} = \frac{1}{2} \cdot 2^n$

$c = \frac{1}{2}$
$n_0 = 1$

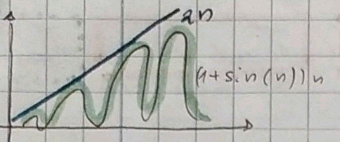
$\boxed{\log n! = O(n \log n)}$ $\log n! = \log \prod_{i=1}^{n} i = \sum_{i=1}^{n} \log i = \underbrace{\log 1}_{\leq n} + \_ + \underbrace{\log n}_{\leq n} = n \cdot \log n$
$c = 1$ $n_0 = 1$

- $(1 + \text{sen}(n))\, n = O(n)$

$\exists c > 0 \quad \exists n_0 \in \mathbb{N} \quad \ni' \quad \forall n \geq n_0 : \quad 1 + \text{sen}(n)\, n \leq cn$ ⤳ per definizione so che $-1 \leq \text{sen}\, n \leq 1$ (sommo 1)

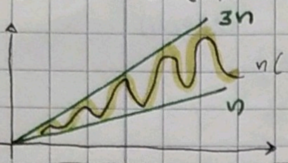
$1 + \text{sen}(n) \leq c$ $\qquad 0 \leq 1 + \text{sen}\, n \leq 2 \quad (c = 2\,!)$

$0 \leq n(1 + \sin n) \leq 2n$



⚠ $\lim_{n \to \infty} \frac{n(1 + \text{sen}(n))}{n} = \nexists$

- $(2 + \sin(n))n = \Theta(n)$



$-1 \leq \sin(n) \leq 1$ (sommo 2)

$1 \leq 2 + \sin(n) \leq 3$ (moltiplico per n) ⤳ ⚠ $\lim_{n \to \infty} \frac{n(2 + \sin(n))}{n} = \nexists$

$n \leq n(2 + \sin(n)) \leq 3n \quad \forall n$

$f(n) = 7n^3 + 5n^2 + 6n + 8$ ⤳ $\lim_{n \to \infty} \frac{7n^3 + 5n^2 + 6n + 8}{n^3} = \lim_{n \to \infty} \left(7 + \frac{5}{n} + \frac{6}{n^2} + \frac{8}{n^3}\right) = 7 \Rightarrow f(n) = \Theta(g(n))$

$g(n) = n^3$

- REGOLA dei POLINOMI: $P(n)$ polinomio (NON negativo) di grado $k \geq 0 \Rightarrow \boxed{P(n) = \Theta(n^k)}$

- BOH, UNA PROPRIETÀ UTILE

$g(n) = f(n) + h(n)$ dove $h(n) = o(f(n)) \Rightarrow \boxed{\begin{array}{l} g(n) = f(n) + o(f(n)) \\ g(n) = \Theta(f(n)) \end{array}}$

dim $\lim_{n \to \infty} \frac{f(n) + h(n)}{f(n)} = \lim_{n \to \infty} \left(1 + \frac{h(n)}{f(n)}^{*}\right) = 1 \in \mathbb{R}$

* per ipotesi $h(n) = o(f(n))$ quindi $\lim_{n \to \infty} \frac{h(n)}{f(n)} = 0$

- $n^2 \log\log n + n^3\sqrt{n} + \log\sqrt{n^2 + 75} \Rightarrow \text{è } \Theta(n^3\sqrt{n})$ ← domina sugli altri termini

$n^{1+\varepsilon} \neq O(n\log\log n) \quad \forall \varepsilon > 0$

$n \log\log n = o(n^{1+\varepsilon})$ ⟶ $\lim_{n \to \infty} \frac{n \log\log n}{n^{1+\varepsilon}} = \lim_{n \to \infty} \frac{\cancel{n} \log\log n}{\cancel{n} \cdot n^{\varepsilon}} \overset{\text{de l'Hopital}}{=} \ldots = 0 \quad \left[\frac{\infty}{\infty}\right]$

- $3n \log n \overset{?}{=} O(5n + \log n^3)$ **NOPE!**

⤳ devo vedere se $\exists c > 0, \exists n_0 \in \mathbb{N} \ni' \forall n \geq n_0 : 3n \log n \leq c(5n + \log n^3)$

quindi mi chiedo $\frac{3n \log n}{5n + \log n^3} \leq c$ $\quad\nexists$ un c che soddisfi la condizione perché il rapporto tende ad infinito

allora passo al limite! $\lim_{n \to \infty} \frac{3n\log n}{5n + \log n^3} = \ldots = +\infty \Rightarrow 3n\log n = \omega(5n + \log n^3)$

$\Rightarrow 3n \log n = \Omega(5n + \log n^3)$

- $2^n = \Omega(2^{n+k})$

$\exists c > 0, \exists n_0 \in \mathbb{N} \ni' \forall n \geq n_0 : \quad 2^n \geq (2^{n+k})\, c \iff$

$\cancel{2^n} \geq \cancel{2^n} \cdot 2^k \cdot c \iff$

$1 \geq c \cdot 2^k \iff$

$c \leq \frac{1}{2^k}$ ⤳ prendo $c = \frac{1}{2^k} \Rightarrow$ è soddisfatta $\forall n_0 \in \mathbb{N}$

$2^{n+n} \overset{?}{=} O(2^n)$ NOPE

$\exists c > 0, \exists n_0 \in \mathbb{N} \ni' \forall n > n_0 : \quad 2^{n+n} \leq c(2^n) \iff$

$2^n \cdot \cancel{2^n} \leq c \cdot \cancel{2^n} \iff$

$2^n \leq c$ ⤳ $\lim_{n \to \infty} 2^n = \infty$

è una successione DIVERGENTE (tende a ∞) quindi $\nexists c$

· $T(n)$ = complessità TEMPORALE dell'algoritmo ~> worst case analysist
- n: dimensione dell' ISTANZA d'ingresso

$A = \{a_1, a_2, \ldots, a_n\}$ -> A | a_1 | a_2 | ... | a_n |
x ... $x \in A$??? myAlgorithm (A, x) ->

$\boxed{T_{best}(n) = const}$ x è in 1° posizione

$\boxed{T_{worst}(n) = n}$ x è alla fine ; $x \notin A$

$\boxed{T_{avg}(n) = \frac{n}{2}}$

i. SEQUENZA

$A \begin{cases} \text{BLOCK\_1} : O(f(n)) \\ \text{BLOCK\_2} : O(g(n)) \end{cases}$

$\Rightarrow O(f(n) + g(n)) : T(n)$

ii. IF THEN ELSE -> $O(f(n))$

if <condizione> then
- ramo_then : $O(g(n))$

else
- ramo_else : $O(h(n))$

~> $T(n) = O(f(n) + \max\{g(n), h(n)\})$
test condizione! worst case!

iii. CICLI

for i=1 to k do
- BLOCK_FOR : $O(g(n))$

~> $T(n) = (k \cdot g(n))$

for i=1 to k do
for j=1 to z do
BLOCK_FOR_FOR : $O(g(n))$

~> $T(n) = (k \cdot z \cdot g(n))$

while <condizione> do -> $O(f(n))$
- BLOCK_WHILE : $O(g(n))$

~> $T(n) = O[\boxed{(N(n))} \cdot (f(n) + g(n))]$
-> max numero di interazioni!!!

iv. CICLO INTERNO CHE DIPENDE DA QUELLO ESTERNO

for i=1 to k
for j=1 to m_i — dipende dall'indice del ciclo esterno ~> $T(n) = m_1 f(n) + m_2 f(n) + \ldots + m_k f(n) = \left(\sum_{i=1}^{k} m_i\right) f(n)$
BLOCK_FOR : $O(f(n))$

· myAlgorithm (int n) -> int
int a, i, j
if (n > 1) then
a = 0
for i=1 to n do ~ n-volte
for j=1 to n do ~ n-volte
a = a + (i+1)(j+1) ~ $\Theta(1)$
} $n \cdot n \cdot \Theta(1) = \Theta(n^2)$
for i=1 to 16 do ~ 16 volte
a = a + myAlgorithm ($\frac{n}{4}$) ~ $T(\frac{n}{4})$
} $16 \cdot T(\frac{n}{4})$
return a
else
return n-1

$$T(n) = \begin{cases} c & n \leq 1 \\ 16T\left(\frac{n}{4}\right) + n^2 & n > 1 \end{cases}$$

master theorem

$T(n) = \Theta(n^2 \log n)$

· metodi per risolvere le ricorrenze

i. ALBERO DELLE RICORSIONI ~> algoritmi semplici
ii. METODO DELL'ITERAZIONE ~> srotolare la ricorrenza e capire cosa fa e come va
iii. METODO DELLA SOSTITUZIONE ~> "intuire" magicamente l'ordine di grandezza e dimostrare per induzione
iv. TEOREMA MASTER ~> « teorema fondamentale delle ricorrenze »

METODO dell' ITERAZIONE

$$T(n) = \begin{cases} 1 & n = 1 \\ c + T\left(\frac{n}{2}\right) & n > 1 \end{cases}$$ algoritmo ricerca binaria

$\leadsto T(n) = 1 \cdot c + T\left(\frac{n}{2}\right) = c + \left[c + T\left(\frac{n}{2^2}\right)\right] = 2 \cdot c + T\left(\frac{n}{2^2}\right) = 2c \left[c + T\left(\frac{n}{2^3}\right)\right] = 3c + T\left(\frac{n}{2^3}\right) = \dots$

assumo che n sia una potenza di 2 per semplicità $= kc + T\left(\frac{n}{2^k}\right)$

⚠ la ricorsione si ferma quando $n = 1 \Rightarrow$ impongo $\frac{n}{2^k} = 1 \sim n = 2^k \leadsto k = \log_2 n$

$\Rightarrow T(n) = c \cdot \log_2 n + T(1) = 1 + c \log_2 n = \boxed{O(\log n)}$

$$T(n) = \begin{cases} 1 & n = 1 \\ 9T\left(\frac{n}{3}\right) + n & n > 1 \end{cases}$$

formula importante
$$\boxed{\sum_{i=0}^{k} q^i = \frac{q^{k+1} - 1}{q - 1}}$$

$\leadsto T(n) = 9T\left(\frac{n}{3}\right) + n = 9\left[9T\left(\frac{n}{3^2}\right) + \frac{n}{3}\right] + n =$

assumo che sia una potenza di 3

$= 9^2\, T\left(\frac{n}{3^2}\right) + 3n + n =$

$= 9^2 \left[9T\left(\frac{n}{3^3}\right) + \frac{n}{3^2}\right] + 3n + n =$

$= 9^3\, T\left(\frac{n}{3^3}\right) + 9n + 3n + n = 9^3\, T\left(\frac{n}{3^3}\right) + n\left(3^2 + 3^1 + 3^0\right) = \dots = 9^k\, T\left(\frac{n}{3^k}\right) + \underbrace{n \sum_{i=0}^{k-1} 3^i}_{n \cdot \frac{3^k - 1}{2}}$

la ricorsione si ferma quando $\frac{n}{3^k} = 1 \leadsto k = \log_3 n$

$\Rightarrow T(n) = 9^k\, T\left(\frac{n}{3^k}\right) + n \cdot \frac{3^k - 1}{2} = 9^{\log_3 n}\, T(1) + n\, \frac{3^{\log_3 n} - 1}{2} =$

$= 3^{2 \log_3 n} + n \cdot \frac{(n-1)}{2} = 3^{2 \log_3 n} + \frac{n(n-1)}{2} =$

$= 3^{\log_3 n^2} + \frac{n(n-1)}{2} = n^2 + \frac{n(n-1)}{2} = n^2 + \frac{n^2}{2} - \frac{n}{2} = \boxed{\Theta(n^2)}$

METODO di SOSTITUZIONE

$$T(n) = \begin{cases} 1 & n = 1 \\ T\left(\left\lfloor \frac{n}{2} \right\rfloor^{*}\right) + n & n \geq 2 \end{cases}$$

* parte intera inferiore $\forall x \in \mathbb{R} : \lfloor x \rfloor \leq x \leq \lceil x \rceil$

i. assumo che $T(n) = O(n)$

ii. dimostro per induzione che ho ragione $\Rightarrow \exists c > 0 \; \exists n_0 \in \mathbb{N} \ni' \forall n \geq n_0 : T(n) \leq c \cdot n$

dim

- CASO BASE $n = 1$: $T(1) = 1 \leq c \cdot 1 \quad \forall c \geq 1$!
- PASSO INDUTTIVO $n > 1$ assumo che $T(n) \leq c \cdot n$ sia vera fino a $n-1$, allora varrà anche per n

$T(n) = T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n$ [per definizione]

$\leq c \cdot \left\lfloor \frac{n}{2} \right\rfloor + n$ [per ip. induttiva ($\left\lfloor \frac{n}{2} \right\rfloor \leq n-1$ allora posso applicarla)]

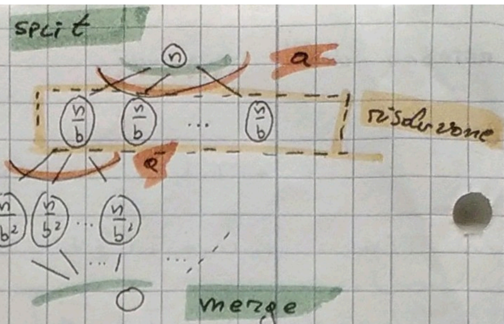
$\leq c \cdot \frac{n}{2} + n$ [$\lfloor x \rfloor \leq x$]

$= \underbrace{\left(\frac{c}{2} + 1\right) n}_{T(n)} \leq c \cdot n \iff 1 \leq \frac{c}{2} \iff c \geq 2 \Rightarrow T(n) \leq c \cdot n$

- $c = 2$
- $n_0 = 1$

• TEOREMA MASTER « divide et impera »

i. SPLIT : scompone il problema in sottoproblemi
ii. RISOLUZIONE : risolvere indipendentemente i sottoproblemi
iii. MERGE : compone la soluzione finale



$T(n) = T_{split}(n) + a\,T\left(\frac{n}{b}\right) + T_{merge}(n) =$
tempo risolvere il problema

$= a\,T\left(\frac{n}{b}\right) + f(n)$

⚠ si assume che ci siano a sottoproblemi di dimensione $\frac{n}{b}$ AD OGNI chiamata ricorsiva ⚠

• $f(n) \geq 0$ * • $a \geq 1$ • $b > 1$ •

* ovviamente per n sufficientemente grande

---

i. $a \geq 1$
ii. $b > 1$
iii. $f(n) \geq 0$
iv. $n > 1$

$\boxed{d = \log_b a}$

$T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$

ordine di grandezza albero ricorsioni

voglio confrontare ASINTOTICAMENTE $f(n)$ e $g(n) = n^d$ perché, intuitivamente, aggiungendo una fz positive ($f(n)$) alla ricorrenza, in alcuni casi, è possibile determinare il comportamento ASINTOTICO di T confrontando f e n^d

v. $\varepsilon > 0$ (deve esistere)

$f(n)$:

- $O(n^{d-\varepsilon})$ $\Rightarrow$ • $T(n) = \Theta(n^d)$ • $f(n)$ cresce MENO velocemente di $g(n)$
- $\Theta(n^d)$ $\Rightarrow$ • $T(n) = \Theta(n^d \log n)$ • $f(n)$ e $g(n)$ crescono in modo simile
- $\Omega(n^{d+\varepsilon})$ + condizione ausiliaria : $\exists\, c < 1$ ∋' per n suff. grande $\left[a f\left(\frac{n}{b}\right) \leq c \cdot f(n)\right]$ $\Rightarrow$ • $T(n) = \Theta(f(n))$ • $f(n)$ cresce più velocemente di $g(n)$

---

boh! esercizi a caso (dopo c'è la dimostrazione del teorema)

• $T(n) = T\left(\frac{n}{2}\right) + c$

condizioni : $a = 1$; $b = 2$; $f(n) = c \geq 0$; ok $\quad d = \log_2 1 = 0 \quad f(n) = c \wedge g(n) = n^d = n^0 = 1$

confronto : $f(n) = c$
$g(n) = n^d = 1 \Rightarrow f(n) = \Theta(n^0) \Rightarrow T(n) = \Theta(n^0 \log n) = \Theta(\log n)$

• $T(n) = 9T\left(\frac{n}{3}\right) + n$

condizioni : $a = 9$; $b = 3$; $f(n) = n \geq 0$ ok $\quad d = \log_3 9 = 2 \quad f(n) = n \wedge g(n) = n^d = n^2$

confronto : $f(n) = n$
$g(n) = n^2$ ~> potrei essere nel 1° caso!

mi chiedo se $f(n) = O(n^{2-\varepsilon})$???
$n = O(n^{2-\varepsilon}) \rightarrow [\varepsilon > 0]$ fisso $\varepsilon = 1 \Rightarrow n = O(n^1) \Rightarrow \boxed{T(n) = \Theta(n^2)}$

• $T(n) = 3T\left(\frac{n}{9}\right) + 1 \quad n \geq 2$

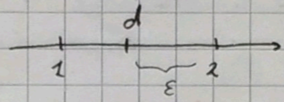
condizioni : $a = 3$; $b = 9$; $1 \geq 0$ ok; $\quad d = \log_9 3 = \frac{1}{2} \quad f(n) = 1 \wedge g(n) = n^{\frac{1}{2}} = \sqrt{n}$

confronto : $f(n) = 1$
$g(n) = \sqrt{n} = n^{\frac{1}{2}}$

mi chiedo se $1 = O(n^{\frac{1}{2}-\varepsilon})$? ~> fisso $\varepsilon = \frac{1}{2} \Rightarrow 1 = O(1) \Rightarrow \boxed{T(n) = \Theta(\sqrt{n})}$

- $T(n) = 3T\left(\frac{n}{2}\right) + n^2$

  $a = 3$; $b = 2$; $d = \log_2 3$ $\begin{cases} 2 < 3 < 4 \\ \log_2 2 < \log_2 3 < \log_2 4 \\ 1 < \log_2 3 < 2 \end{cases}$



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

$f(n) = n^2$; $g(n) = n^d$

- $n^2 = \Omega(n^{d+\varepsilon})$ -> impongo $d + \varepsilon \leq 2 \Rightarrow \log_2 3 + \varepsilon \leq 2 \leadsto 0 < \varepsilon \leq 2 - \log_2 3$

- condizione ausiliaria -> $\exists\, c < 1$ t.c. p.n.s.g. $a f\left(\frac{n}{b}\right) \leq c \cdot f(n)$

  $3\left(\frac{n}{2}\right)^2 \leq c \cdot n^2$; $\frac{3\cancel{n^2}}{4} \leq c \cdot \cancel{n^2}$; $3/4 \leq c < 1$

  $c = 3/4$ verifica la condizione

- $T(n) = 4T\left(\frac{n}{2}\right) + n^2$

  $a = 4$; $b = 2$, $d = \log_b a = \log_2 4 = 2$ $f(n) = n^2$ $g(n) = n^d = n^2$

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

- $T(n) = 2^n\, T\left(\frac{n}{2}\right) + n^2$: $a = 2^n$ **NON** è costante
- $T(n) = 0.5\, T\left(\frac{n}{2}\right) + \frac{1}{n}$: $a = 0.5 < 1$
- $T(n) = 4\, T\left(\frac{n}{3}\right) + \sin(n)$: p.n.s.g. $\sin(n) \not> 0$

⚠ **NON** si può applicare il teorema master ⚠

- $T(n) = 3T\left(\frac{n}{4}\right) + n\log n$

  $a = 3$; $b = 4$; $n \log n > 0$ p.n.s.g. ; $d = \log_4 3 \leadsto \frac{1}{2} < \log_4 3 < 1$

  $f(n) = n \log n$
  $g(n) = n^d = n^{\log_4 3}$

- $f(n) = \Omega(n^{d+\varepsilon}) \leadsto n\log n = \Omega(n^{d+\varepsilon})$

  impongo $d + \varepsilon \leq 1$
  $\log_4 3 + \varepsilon = 1$
  $1 - \log_4 3 = \varepsilon$

- CONDIZIONE AUX

  $\exists\, c < 1$ t.c. p.n.s.g. $a f\left(\frac{n}{b}\right) \leq c \cdot f(n)$ -> $3\left(\frac{n \log\left(\frac{n}{4}\right)}{4}\right) \leq c \cdot n \log n$

  $3/4 \log \frac{n}{4} \leq c \cdot \log n$

  $3/4\left(\log_4 n - \cancel{\log_4 4}^{\,1}\right) \leq c \cdot \log n$

  $3/4 \log_4 n - 3/4 \leq c \cdot \log n$

  $(3/4 - c) \log n \leq 3/4$ boh ¨

  impongo che $\frac{3}{4} - c = 0 \Rightarrow c = 3/4$

  ⇓

  $\boxed{T(n) = \Theta(n \log n)}$

- $T(n) = 2T\left(\frac{n}{2}\right) + n\log n$

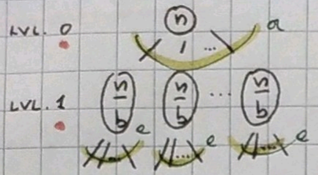
  ci sono tutte le condizioni per poter applicare il teorema master

  MA, non sono in nessuno dei 3 casi ¨

  (spero NON capiti all' esame)

· Dimostrazione t. master :

: scrivere $T(n)$ in maniera [redacted] (NON ricorsiva) ESPLICITA



- numero di nodi a livello $i\ [i \geq 0] = a^i$
- dimensionalità del problema a livello $i : \frac{n}{b^i}$
- contributo delle chiamate a livello $i : f\left(\frac{n}{b^i}\right)$

$$T(n) = \sum_{i=0}^{\log_b n} \underbrace{a^i f\left(\frac{n}{b^i}\right)}_{\text{costo livello i-esimo!}}$$

⇝ [la ricorsione si ferma quando $\frac{n}{b^i} = 1 \iff n = b^i \iff i = \log_b n$] numero di livelli dell'albero!

▸ NUMERO di FOGLIE

$$a^i = a^{\log_b n} \underset{\text{cambio base}}{=} a^{(\log_a n \cdot \log_b a)} = \left(a^{\log_a n}\right)^{\log_b a} = n^{\log_b a} = n^d$$

rappresenta anche l'ordine di grandezza dell'albero delle ricorsioni

▸ NUMERO TOTALE di CHIAMATE / VERTICI / NODI dell'ALBERO

$$\sum_{i=0}^{\log_b n} a^i = \frac{a^{\log_b n+1} - 1}{a-1} = \frac{a^{\log_b n} \cdot a - 1}{a-1} = \frac{n^d \cdot \cancel{a-1}}{\cancel{a-1}} = \boxed{\Theta(n^d)}$$

costanti, le ignoro

→ cresce come n^d, asintoticamente!

· dim I° caso !

- IPOTESI : $f(n) = O(n^{d-\varepsilon})$
- TESI : $T(n) = \Theta(n^d) \Rightarrow \underset{(i)}{T(n) = O(n^d)} \wedge \underset{(ii)}{T(n) = \Omega(n^d)}$

$$a^i f\left(\frac{n}{b^i}\right) = a^i O\left(\left(\frac{n}{b^i}\right)^{d-\varepsilon}\right) = \quad [\text{per } i = 0 \dots \log_b n]$$

$$* \quad (b^d)^i = (b^{\log_b a})^i = a^i$$

ho scambiato gli esponenti

$$= O\left(\cancel{a^i} \frac{n^{d-\varepsilon}}{\cancel{(b^d)^i}\,(b^{-\varepsilon})^i}\right) = O(n^{d-\varepsilon} \cdot (b^\varepsilon)^i)$$

risolvo $T(n)$ ora

$$(i)\ T(n) = \sum_{i=0}^{\log_b n} a^i f\left(\frac{n}{b^i}\right) = \sum_{i=0}^{\log_b n} O(n^{d-\varepsilon} \cdot (b^\varepsilon)^i) = O\left[\sum_{i=0}^{\log_b n} n^{d-\varepsilon} \cdot (b^\varepsilon)^i\right] =$$

$$= O\left(n^{d-\varepsilon} \cdot \sum_{i=0}^{\log_b n} (b^\varepsilon)^i\right) = O\left(n^{d-\varepsilon} \cdot \frac{(b^\varepsilon)^{\log_b n+1} - 1}{b^\varepsilon - 1}\right) = O\left(n^{d-\varepsilon} \cdot \frac{(b^{\log_b n})^\varepsilon \cdot b^\varepsilon - 1}{b^\varepsilon - 1}\right) =$$

sono costanti!

$$= O\left(n^{d-\varepsilon} \cdot n^\varepsilon \frac{\cancel{b^\varepsilon - 1}}{\cancel{b^\varepsilon - 1}}\right) = O(n^{d-\varepsilon} \cdot n^\varepsilon) = O(n^d)$$

numero di foglie, quindi n^d ALMENO dovrò farle!

(ii) $T(n) = \Omega(n^d)$ ⇝ $T(n) \geq n^d$ ma allora $T(n) = O(n^d) \wedge T(n) = \Omega(n^d) \Rightarrow \boxed{T(n) = \Theta(n^d)}$

· dim II° caso !

- IPOTESI : $f(n) = \Theta(n^d)$
- TESI : $T(n) = \Theta(n^d \log n)$

⚠ il costo del livello i-esimo NON dipende da i !

$$a^i f\left(\frac{n}{b^i}\right) = \Theta\left(a^i \left(\frac{n}{b^i}\right)^d\right) = \Theta\left(\cancel{a^i} \cdot \frac{n^d}{\cancel{(b^d)^i}}\right) = \Theta(n^d)$$

perché inizio da zero !

$$T(n) = \sum_{i=0}^{\log_b n} a^i f\left(\frac{n}{b^i}\right) = \sum_{i=0}^{\log_b n} \Theta(n^d) = \Theta\left(\sum_{i=0}^{\log_b n} n^d\right) = \Theta(n^d(\log_b n + 1)) =$$

$$= \Theta(\underbrace{n^d \log n}_{\text{domina rispetto } n^d} + \cancel{n^d}) = \boxed{\Theta(n^d \log n)}$$

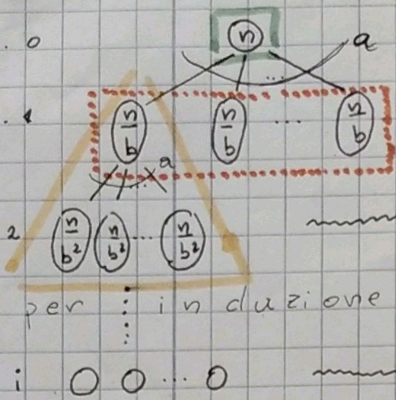
numero livelli dell'albero!

complessità livello (ognuno costa n^d)

dim III° caso!

- IPOTESI 1: $f(n) = \Omega(n^{d+\varepsilon})$ $\quad \varepsilon > 0$
- IPOTESI 2: $\exists c<1$ t.c. p.n.s.g. $\underbrace{a f\left(\frac{n}{b}\right)}_{\text{tempo complessivo al I° livello}} \le \underbrace{c \cdot f(n)}_{\text{tempo alla radice (split + merge) alla I}^a\text{ chiamata ricorsiva}}$

si perde più tempo nello split e merge rispetto alle chiamate ricorsive!!!



LVL. 1 ⇝ $a f\left(\frac{n}{b}\right) \le c \cdot f(n)$ vale per p.n.s.g. quindi anche per $\frac{n}{b^2}$

LVL. 2 ⇝ $a f\left(\frac{n}{b^2}\right) \le c \cdot f\left(\frac{n}{b}\right)$ ⇝ $a^2 f\left(\frac{n}{b^2}\right) \le c \cdot c f(n) = \boxed{c^2 f(n)}$

⚠ a un livello i-esimo dell'albero perderò MENO tempo rispetto a quello alla radice ⚠

LVL. i ⇝ $a^i f\left(\frac{n}{b^i}\right) \le c^i f(n)$

$$T(n) = \sum_{i=0}^{\log_b n} a^i f\left(\frac{n}{b^i}\right) \le \sum_{i=0}^{\log_b n} c^i f(n) = \underbrace{f(n) \sum_{i=0}^{\log_b n} c^i}_{\text{SOMMA FINITA}} \le \underbrace{f(n) \sum_{i=0}^{\infty} c^i}_{\text{serie geometrica convergente}} = f(n)\left(\frac{1}{1-c}\right) = O(f(n))$$

boh, posso farlo perché una somma di termini POSITIVI sarà per forza $\le$ di una somma di termini INFINITA

Serie geometrica

$q < 1$

$$\sum_{i=0}^{\infty} q^i = \frac{1}{1-q}$$

convergente

$$T(n) = \underbrace{a T\left(\frac{n}{b}\right)}_{\text{quantità } \ge 0} + f(n) \quad \text{[per definizione]} \Rightarrow T(n) \ge f(n) \leadsto \boxed{T(n) = \Theta(f(n))}$$