# CLASSI COMPLESSITÀ

- 1° CLASSE COMPLESSITÀ [O omicron] : limite superiore

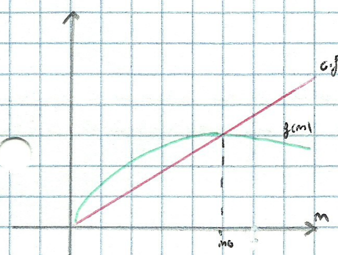
$$O(g(n)) = \{ \exists\, c>0 \wedge \exists\, n_0 \in \mathbb{N} \mid \forall n \geq n_0 : f(n) \leq c \cdot g(n) \}$$

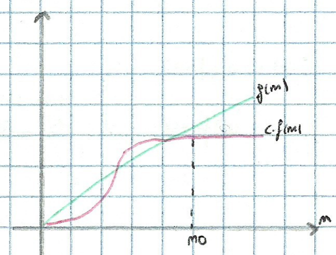
- 2° CLASSE COMPLESSITÀ [$\Omega$ omega] : limite inferiore

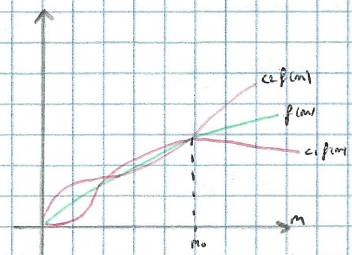
$$\Omega(g(n)) = \{ \exists\, c>0 \wedge n_0 \in \mathbb{N} \mid \forall n \geq n_0 : f(n) \geq c \cdot g(n) \}$$

- 3° CLASSE COMPLESSITÀ [$\Theta$ teta] : limite stretto

$$\Theta(g(n)) = \{ \exists\, c_1, c_2 > 0 \wedge \exists\, n_0 \in \mathbb{N} \mid \forall n \geq n_0 : c_1 g(n) \leq f(x) \leq c_2 \cdot f(x) \}$$




# CLASSI COMPLESSITÀ ASINTOTICA

- $T(n) = O(1)$ ⇒ complessità costante, cioè non dipende dalla dimensione n dei dati in INPUT
- $T(n) = O(\log n)$ ⇒ complessità logaritmica
- $T(n) = O(n)$ ⇒ complessità lineare
- $T(n) = O(n \log n)$ ⇒ complessità pseudolineare (cioè detto da $n \log n = O(n^{1+\varepsilon})$ per ogni $\varepsilon > 0$)
- $T(n) = O(n^2)$ ⇒ complessità quadratica
- $T(n) = O(n^3)$ ⇒ complessità cubica
- $T(n) = O(n^k)$ ⇒ complessità polinomiale (con $k>0$)
- $T(n) = O(a^n)$ ⇒ complessità esponenziale (con $a > 1$)

# SOTTOCLASSI COMPLESSITÀ

[se dimostro che $f(n)$ è $o(g(n))$ oppure $\omega(g(n))$ allora saranno $O(g(n))$ oppure $\Omega(g(n))$]

- CLASSE o [omicron piccolo]

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

$\boxed{NB}$: $o(g(n)) \subseteq O(g(n))$

- CLASSE $\omega$ [omega piccolo]

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

$\boxed{NB}$: $\omega(g(n)) \subseteq \Omega(g(n))$

$\boxed{NB}$: $o(g(n)) \cap \Omega(g(n)) = \emptyset$

$\omega(g(n)) \cap O(g(n)) = \emptyset$

# PROPRIETÀ COMPLESSITÀ

- PROPRIETÀ TRANSITIVA

$$f(n) \begin{cases} O(g(n)) \\ \Omega(g(n)) \\ \Theta(g(n)) \\ o(g(n)) \\ \omega(g(n)) \end{cases} \wedge \quad g(n) \begin{cases} O(h(n)) \\ \Omega(h(n)) \\ \Theta(h(n)) \\ o(h(n)) \\ \omega(h(n)) \end{cases} = \quad f(n) \begin{cases} O(h(n)) \\ \Omega(h(n)) \\ \Theta(h(n)) \\ o(h(n)) \\ \omega(h(n)) \end{cases}$$

- PROPRIETÀ RIFLESSIVA

$f(n) = O(f(n))$

$f(n) = \Omega(f(n))$

$f(n) = \Theta(f(n))$

$$\left.\begin{array}{l} f(n) = o(f(n)) \\ f(n) = \omega(f(n)) \end{array}\right\}$$ non vale poiché se la relazione è strettamente maggiore non può essere strettamente minore

# ORDINAMENTO COMPLESSITÀ [dal minore al maggiore]

$1$, $\log\log n$, $\log n$, $\sqrt{n}$, $n$, $n\log n$, $n\sqrt{n}$, $n^2$, $n^2\log n$, $n^3$, ..., $2^n$, ... $n!$, ... $n^n$

es: $\log\log n$, $17\log n$, $\sqrt{n}$, ~~[illegible]~~, $n\log n$, $10n^{3/2}$, $n^2$, $n^2 + 2n\ln n$, $5n^2\log\log n$

$n^3$, $n + \log n$, $3n^2 + n^2\log n$, $n^5 - n^4 + 2n$, ...

# USO MASTER THEOREM e METODI VARI CALCOLO COMPLESSITÀ

[$T(n) = aT(\frac{n}{b}) + f(n)$]

- [1° CASO] $\quad T(n) = 4T(\frac{n}{2}) + n$

  - $n^{\log_b a} = n^{\log_2 4} = n^2$
  - $f(n) = n = n^{2-\varepsilon}$ con $(\varepsilon = 1) > 0$

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

- [2° CASO] $\quad T(n) = 4T(n/2) + n^2$

  - $n^{\log_b a} = n^{\log_2 4} = n^2$
  - $f(n) = n^2$

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

- [3° CASO] $T(n) = 4T(n/2) + n^3$
  - $n^{\log_b a} = n^{\log_2 4} = n^2$
  - $f(n) = n^3$

quindi ricordandoci che $a \cdot f(n) \leq c \cdot f(n)$ per qualche $c < 1$ ed $n$ sufficientemente grande, infatti:

$$4(n/2)^3 = n^3/2 \leq cn^3 \quad \text{se} \quad c \geq 1/2$$

quindi $T(n) = \Theta(n^3)$

- [RICORSIONE] $T(n) = T(n/4) + T(3n/4) + n$
  - al primo livello abbiamo un costo di $cn$ dovuto alla componente $n$ della ricorsione
  - al secondo livello abbiamo $1/4\,cn + 3/4\,cn$, dovuto alla componente $n$ delle sottoparti
  - al terzo livello abbiamo un costo $(1/16\,cn + 3/16\,cn + 3/16\,cn + 9/16\,cn) = cn$ dovuto alla componente $n$ delle quattro sottoparti
  - e così via...

quindi è facile verificare che procederà per almeno $\log_4 n$ livelli dove $n/4^2$ scenderà sotto il valore 1. I livelli successivi saranno limitati superiormente da un costo $cn$, fino al livello $\log_{4/3} n$ quando $3/4^2$ scenderà sotto il valore 1
ovunque:

$$cn \log_4 n \leq T(n) \leq cn \log_{4/3} n \quad \Rightarrow T(n) = \Theta(n \log n)$$

- [ITERAZIONE] $T(n) = T(n-d) + T(d) + n^2$ $\quad d \geq 1$ (costante)
  - $T(d)$ sarà costante
  - $T(n) = T(n-d) + cd + n^2 \xrightarrow{\text{ITERO}}$ $T(n-d) + d + n^2$
    $T(n-2d) + d + (n-d)^2 + d + n^2$
    $T(n-3d) + d + (n-2d)^2 + d + (n-d)^2 + d + n^2$
    ecc...

$$\rightarrow n + \sum_{i=0}^{n/d} (n - id)^2$$

$$n + \sum_{i=0}^{n/d} (n^2 - 2ind + i^2d^2) \rightarrow n + \sum_{i=0}^{n/d} n^2 - \sum_{i=0}^{n/d} 2nid + \sum_{i=0}^{n/d} i^2d^2$$

$$\rightarrow n + n^3/d + n^2 - 2nd\,\frac{((n/d)+1)\,n/d}{2} + d^2 \sum_{i=0}^{n/d} i^2$$

$$\rightarrow n + (n^3/d) + n^2 - (n^3/d) - n^2 + d^2\,\frac{(n/d)(n/d+1)(2n/d+1)}{6}$$

$$\rightarrow T(n) = \Theta(n^3)$$

- [INDUZIONE] $T(n) = T(n-1) + \log n$

  - Dimostriamo che $T(n) = O(n \log n)$
  - poniamo $T(m) \leq cm \log m$ per tutti gli $m < n$
  - dimostriamo che $n$ è vero: 
  $$\begin{aligned} T(n) &= T(n-1) + \log n \\ &\leq c(n-1)\log(n-1) + \log n \\ &\leq c(n-1)\log n + \log n \\ &= cn\log n - c\log n + \log n \end{aligned}$$
  - Abbiamo che $cn\log n - c\log n + \log n \leq cn\log n$ se $c \geq 1$
  - CASO BASE $\rightarrow$ non può essere utilizzato il valore 1 essendo $\log 1 = 0$, ma prendendo $T(2) = T(1) + \log 2 \rightarrow 1+1 \leq 2c\log 2 = 2c$

    CVD

- [SOSTITUZIONE] $T(n) = 2T(n^{1/2}) + \log n$

  - poniamo $n = 2^m \rightarrow T(2^m) = 2T(2^{m/2}) + \log 2^m$
  - poniamo $S(m) = T(2^m) \rightarrow S(m) = 2S(m/2) + m$
  - $S(m) = \Theta(m \log m) \rightarrow T(n) = \Theta(\log n \log\log n)$

- [SOSTITUZIONE] $T(n) = \sqrt{n}\, T(\sqrt{n}) + n \log \sqrt{n}$

  - poniamo $n = 2^k$ (oppure $k = \log n$)
  - $T(2^k) = 2^{k/2} T(2^{k/2}) + 2^k \log 2^{k/2} \rightarrow 2^{k/2} T(2^{k/2}) + 2^k\, k/2$
  - eliminando $2^k \rightarrow \dfrac{T(2^k)}{2^k} = \dfrac{T(2^{k/2})}{2^{k/2}} + \dfrac{k}{2}$
  - $S(k) = \dfrac{T(2^k)}{2^k} \Rightarrow S(k) = S(k/2) + k/2$
  - uso Master Theorem $S(k) = \Theta(k)$
  - $\Rightarrow \dfrac{T(2^k)}{2^k} = \Theta(k) \rightarrow T(2^k) = \Theta(k)2^k \rightarrow T(n) = \Theta(\log n)n$

    $\rightarrow T(n) = \Theta(n \log n)$

- [SOSTITUZIONE]

$$T \begin{cases} 1 & \text{se } n = 1 \\ 2T(n/3) + T(2n/3) + 1 & n > 1 \end{cases}$$

lim sup? lim inf?

  - $T(n) = 2T(n/3) + T(2n/3) + 1 \geq 2T(n/3) + T(n/3) + 1 = 3T(n/3) + 1$

    e

    $T(n) = 2T(n/3) + T(2n/3) + 1 \leq 2T(2n/3) + T(2n/3) + 1 = 3T(2n/3) + 1$
  - Applicando il Master Theorem: $T(n) \geq 3T(n/3) + 1 = \Theta(n)$

    e

    $T(n) \leq 3T(2n/3) + 1 = \Theta(n^{\log_{3/2} 3}) \Rightarrow \Theta(n$

Quindi è possibile dimostrare un limite stretto usando $T(n) = O(n^2)$ per sostituzione. Nel caso generale esistono $c > 0$, $m > 0$ tali per cui $T(n) \leq cn^2$ per ogni $n \geq m$

$$T(n) = 2T(n/3) + T(2n/3) + 1$$
$$\leq 2cn^2/9 + 4cn^2/9 + 1$$
$$= 2/3\,cn^2 + 1 \leq cn^2 \quad \text{se } n \geq m = 1, \; c \geq 3$$

- caso base: $T(1) = 1 \leq c \cdot 1^2$ vero se $c \geq 1$

# MASTER THEOREM

[GENERALE] Sia $a \geq 1$, $b > 1$, $\begin{cases} T(n) = aT(n/b) + f(n) \\ n \geq 1 \end{cases}$

- 1° CASO: se $f(n) = O(n^{d-\varepsilon})$ per $\varepsilon > 0$

  allora $T(n) = \Theta(n^d)$

- 2° CASO: se $f(n) = \Theta(n^d)$

  allora $T(n) = \Theta(n^d \lg n)$

- 3° CASO: se $f(n) = \Omega(n^{d+\varepsilon})$ per $\varepsilon > 0$

  e se $af(n/b) \leq c \cdot f(n)$ per $c < 1$ e per $n$ sufficientemente grande

  allora $T(n) = \Theta(f(n))$