UNITELMA SAPIENZA

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

# Corso di Introduzione agli algoritmi
Prof.ssa Tiziana Calamoneri

# Notazione asintotica

UNITELMA SAPIENZA
SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

A cosa serve il costo computazionale?

- Per stimare il tempo impiegato da un algoritmo
- Per confrontare l'efficienza di algoritmi diversi
- Per stimare la più grande dimensione dell'input gestibile in tempi ragionevoli

Il costo computazionale ha senso quando la dimensione dell'input è sufficientemente grande

$\Downarrow$

efficienza asintotica

# Notazione *O* (1)

**Definizione**. (limite asintotico superore)
Date due funzioni $f(n), g(n) \geq 0$ si dice che $f(n)$ è un $O(g(n))$ se esistono due costanti $c$ ed $n_0$ tali che $0 \leq f(n) \leq c\, g(n)$ per ogni $n \geq n_0$.

## Notazione O (2)

Esempio. $f(n) = 3n + 3$

$f(n)$ è un $O(n^2)$ infatti:

posto $c = 6$, $cn^2 \geq 3n + 3$ per ogni $n \geq 1$.

$f(n)$ è anche un $O(n)$ infatti:

$cn \geq 3n + 3$ per ogni $n \geq 1$ se $c \geq 6$,

oppure per ogni $n \geq 3$ se $c \geq 4$.

In effetti, data una funzione $f(n)$:

esistono infinite funzioni $g(n)$ per cui $f(n)$ risulta un $O(g(n))$.

Vogliamo determinare la funzione $g(n)$ che meglio approssima la funzione $f(n)$ dall'alto o, informalmente, la più piccola funzione $g(n)$ tale che $f(n)$ sia $O(g(n))$.

**Regola**. Sia *f(n)* un polinomio di grado *m*:

$$f(n)=\sum_{i=0}^{m} a_i n^i \quad \text{con } a_m > 0,$$

allora *f(n)* è un $O(n^m)$.

**Dim**. per induzione su *m* aggiungendo l'ulteriore proprietà che

$$c \geq \sum_{i=0}^{m} a_i$$

**Casi base:**

***m* = 0**: $f(n) = a_0$, quindi *f(n)* è una costante, cioè un *O(1)* per ogni *n* e per ogni $c \geq a_0$.

***m* = 1**: $f(n) = a_0 + a_1 n$, quindi *f(n)* è un *O(n)* per qualunque *n*, per ogni $c \geq a_0 + a_1$

## Notazione O (4)

**Segue dim. Regola**. Un polinomio di grado *m* $f(n)=\sum_{i=0}^{m} a_i n^i$ con $a_m > 0$, è un $O(n^m)$.

**Ipotesi induttiva:**

$h(n)=\sum_{i=0}^{m-1} a_i n^i$ è un $O(n^{m-1})$, cioè $cn^{m-1} \geq \sum_{i=0}^{m-1} a_i n^i$ se $c \geq \sum_{i=0}^{m-1} a_i$

**Passo induttivo:**

Dobbiamo dimostrare che $f(n)=\sum_{i=0}^{m} a_i n^i$ è un $O(n^m)$

cioè che $c' n^m \geq \sum_{i=0}^{m} a_i n^i$ con $c' \geq \sum_{i=0}^{m} a_i$

$f(n)$ si può scrivere come $f(n) = h(n) + a_m n^m$

per hp indutt. $cn^{m-1} \geq h(n)$ con $c \geq \sum_{i=0}^{m-1} a_i$

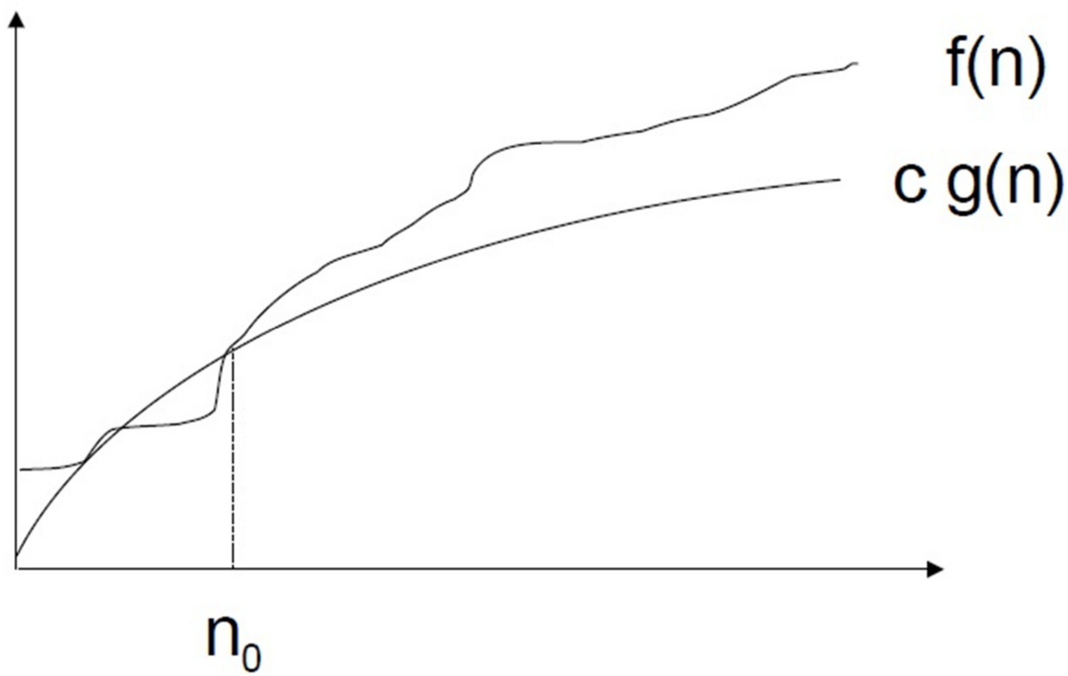
Ora:

$f(n) = h(n) + a_m n^m \leq cn^{m-1} + a_m n^m \leq cn^m + a_m n^m = (c + a_m)n^m$

Ponendo $c' = c + a_m$ si ha la tesi.

CVD

**Definizione**. (limite asintotico inferiore)
Date due funzioni $f(n)$, $g(n) \geq 0$ si dice che $f(n)$ è un $\Omega(g(n))$ se esistono due costanti $c$ ed $n_0$ tali che $f(n) \geq c\, g(n)$ per ogni $n \geq n_0$.



## Notazione Ω (2)

Esempio. $f(n) = 2n^2 + 3$

$f(n)$ è un $\Omega(n)$ infatti:

posto $c = 1$, $2n^2 + 3 \geq cn$ per qualunque $n$ .

$f(n)$ è anche un $\Omega(n^2)$ infatti:

$2n^2 + 3 \geq cn^2$ per ogni $n$, se $c \leq 2$.

Anche in questo caso, data una funzione $f(n)$:

esistono molte funzioni $g(n)$ per cui $f(n)$ risulta un $\Omega(g(n))$.

Vogliamo determinare la funzione $g(n)$ che meglio approssima la funzione $f(n)$ dal basso o, informalmente, la più grande funzione $g(n)$ tale che $f(n)$ sia $\Omega(g(n))$.

## Notazione Ω (3)

**Regola**. Sia *f(n)* un polinomio di grado *m*:

$$f(n)=\sum_{i=0}^{m} a_i n^i \quad \text{con } a_m > 0,$$

allora *f(n)* è un $\Omega(n^m)$.

**Dim**. Per esercizio

## Notazione Ω (4)

UNITELMA SAPIENZA
SAPIENZA Università di Roma
Dipartimento di Informatica

- In entrambe le notazioni appena esposte, per ogni funzione *f(n)* è possibile trovare più funzioni *g(n)*.
- In effetti *O(g(n))* e *Ω(g(n))* sono ***insiemi di funzioni***, e dire:

“*f(n)* è un *O(g(n)*” oppure “*f(n)* = *O(g(n)”* ha il significato di

“*f(n)* appartiene a *O(g(n))*”.

- Poiché i limiti asintotici ci servono per stimare con la maggior precisione possibile il costo computazionale di un algoritmo, vorremmo trovare – fra tutte le possibili funzioni *g(n)* – quella che più si avvicina a *f(n)*.
- Per questo cerchiamo la più piccola funzione *g(n)* per determinare *O* e la più grande funzione *g(n)* per determinare *Ω*.

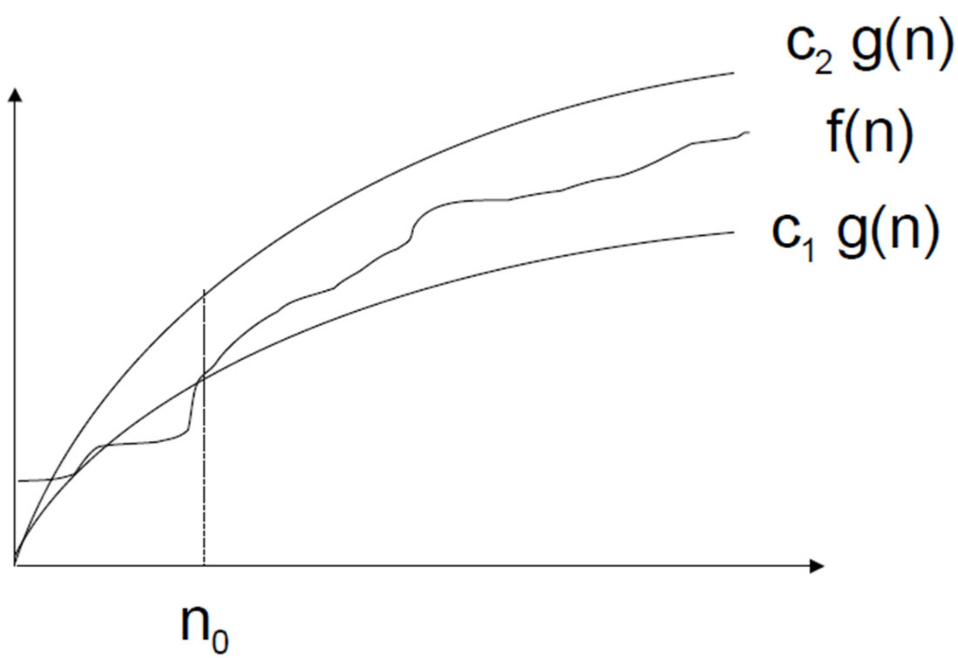
## Notazione Θ (1)

**Definizione**. (limite asintotico stretto)

Date due funzioni $f(n), g(n) \geq 0$ si dice che $f(n)$ è un $\Theta(g(n))$ se esistono tre costanti $c_1$, $c_2$ ed $n_0$ tali che

$$c_1\, g(n) \leq f(n) \leq c_2\, g(n) \text{ per ogni } n \geq n_0.$$

In altre parole, $f(n)$ è $\Theta(g(n))$ se è contemporaneamente $O(g(n))$ e $\Omega(g(n))$.



## Notazione Θ (2)

Esempio. $f(n) = 3n + 3$ è un $\Theta(n)$

Basta porre ad es. $c_1 = 3$, $c_2 = 4$, $n_0 = 3$.

**Regola**. Sia $f(n)$ un polinomio di grado $m$:

$$f(n) = \sum_{i=0}^{m} a_i n^i \text{ con } a_m > 0,$$

allora $f(n)$ è un $\Theta(n^m)$.

## Regole sulle costanti moltiplicative

**1A**: Per ogni $k > 0$ e per ogni $f(n) \geq 0$, se $f(n)$ è un $O(g(n))$ allora anche $k\,f(n)$ è un $O(g(n))$.

**1B**: Per ogni $k > 0$ e per ogni $f(n) \geq 0$, se $f(n)$ è un $\Omega(g(n))$ allora anche $k\,f(n)$ è un $\Omega(g(n))$.

**1C**: Per ogni $k > 0$ e per ogni $f(n) \geq 0$, se $f(n)$ è un $\Theta(g(n))$ allora anche $k\,f(n)$ è un $\Theta(g(n))$.

Informalmente, queste tre regole si possono riformulare dicendo che le costanti moltiplicative si possono ignorare.

## Regole sulla commutatività con la somma

**2A**: Per ogni $f(n), d(n) > 0$, se $f(n)$ è un $O(g(n))$ e $d(n)$ è un $O(h(n))$ allora $f(n)+d(n)$ è un $O(g(n)+h(n)) = O(max(g(n),h(n)))$.

**2B**: Per ogni $f(n), d(n) > 0$, se $f(n)$ è un $\Omega(g(n))$ e $d(n)$ è un $\Omega(h(n))$ allora $f(n)+d(n)$ è un $\Omega(g(n)+h(n)) = \Omega(max(g(n),h(n)))$.

**2C**: Per ogni $f(n), d(n) > 0$, se $f(n)$ è un $\Theta(g(n))$ e $d(n)$ è un $\Theta(h(n))$ allora $f(n)+d(n)$ è un $\Theta(g(n)+h(n)) = \Theta(max(g(n),h(n)))$.

Informalmente, queste tre regole si possono riformulare dicendo che le notazioni asintotiche commutano con l'operazione di somma.

UNITELMA SAPIENZA
SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

## Regole sulla commutatività col prodotto

**3A**: Per ogni *f(n), d(n) > 0*, se *f(n)* è un *O(g(n))* e *d(n)* è un *O(h(n))* allora *f(n)d(n)* è un *O(g(n)h(n))*.

**3B**: Per ogni *f(n), d(n) > 0*, se *f(n)* è un Ω*(g(n))* e *d(n)* è un Ω*(h(n))* allora *f(n)d(n)* è un Ω*(g(n)h(n))*.

**3C**: Per ogni *f(n), d(n) > 0*, se *f(n)* è un Θ*(g(n))* e *d(n)* è un Θ*(h(n))* allora *f(n)d(n)* è un Θ*(g(n)h(n))*.

Informalmente, queste tre regole si possono riformulare dicendo che le notazioni asintotiche commutano con l'operazione di prodotto.

**Esempio**. Trovare il limite asintotico stretto per $f(n) = 3n2^n + 4n^4$

$3n2^n + 4n^4 = \Theta(n)\Theta(2^n) + \Theta(n^4) = \Theta(n2^n) + \Theta(n^4) = \Theta(n2^n).$

**Esempio**. Trovare il limite asintotico stretto per $f(n) = 2^{n+1}$

$2^{n+1} = 2\ 2^n = \Theta(2^n).$

**Esempio.** Trovare il limite asintotico stretto per $f(n) = 2^{2n}$

$2^{2n} = \Theta(2^{2n}).$

**Regola 1A.** Per ogni $k > 0$ e per ogni $f(n) \geq 0$, se $f(n)$ è un $O(g(n))$ allora anche $k\, f(n)$ è un $O(g(n))$.

**Dim.** Per ipotesi, $f(n)$ è un $O(g(n))$ quindi esistono due costanti $c$ ed $n_0$ tali che:

$$f(n) \leq cg(n) \text{ per ogni } n \geq n_0$$

Ne segue che:

$$kf(n) \leq kcg(n)$$

Questo prova che, prendendo $kc$ come nuova costante $c'$ e mantenendo lo stesso $n_0$, $kf(n)$ è un $O(g(n))$. CVD

## Algebra della notaz. asintotica (6)

**Regola 2A.** Per ogni $f(n), d(n) > 0$, se $f(n)$ è un $O(g(n))$ e $d(n)$ è un $O(h(n))$ allora $f(n)+d(n)$ è un $O(g(n)+h(n)) = O(max(g(n),h(n)))$.

**Dim.** Se $f(n)$ è un $O(g(n))$ e $d(n)$ è un $O(h(n))$ allora esistono quattro costanti $c'$ e $c''$, $n'_0$ ed $n''_0$ tali che:

$f(n) \leq c'g(n)$ per ogni $n \geq n'_0$ e $d(n) \leq c''h(n)$ per ogni $n \geq n''_0$

Allora:

$f(n) + d(n) \leq c'g(n) + c''h(n) \leq max(c', c'')(g(n) + h(n))$ per ogni $n \geq max(n'_0, n''_0)$

Segue che $f(n) + d(n)$ è un $O(g(n)+h(n))$.

Infine: $max(c', c'')(g(n) + h(n)) \leq 2\ max(c', c'')\ max(g(n), h(n))$.

Ne segue che $f(n) + d(n)$ è un $O(max(g(n), h(n)))$.

CVD

**Regola 3A.** Per ogni $f(n)$, $d(n) > 0$, se $f(n)$ è un $O(g(n))$ e $d(n)$ è un $O(h(n))$ allora $f(n)d(n)$ è un $O(g(n)h(n))$.

**Dim.** Se $f(n)$ è un $O(g(n))$ e $d(n)$ è un $O(h(n))$ allora esistono quattro costanti $c'$ e $c''$, $n'_0$ ed $n''_0$ tali che:

$f(n) \leq c'g(n)$ per ogni $n \geq n'_0$ e $d(n) \leq c''h(n)$ per ogni $n \geq n''_0$

Allora:

$f(n)d(n) \leq c'c''g(n)h(n)$ per ogni $n \geq max(n'_0, n''_0)$

Da ciò segue che $f(n)d(n)$ è un $O(g(n)h(n))$.

CVD

Fare per esercizio le dimostrazioni delle altre regole che coinvolgono le notazioni $\Omega$ e $\Theta$.

Calcolare l'andamento asintotico delle seguenti funzioni:

- $f(n)=n^2 \log n$
- $f(n)=3n \log n+2n^2$
- $f(n)=2^{\log n/2}+5n$
- $f(n)=4^{\log n}$
- $f(n)=(\sqrt{2})^{\log n}$