Corso di laurea in Informatica
Introduzione agli Algoritmi
Lezioni in modalità mista o a distanza
A.A. 2021/22

Notazione Asintotica

Tiziana Calamoneri

SAPIENZA
UNIVERSITÀ DI ROMA

Slides realizzate sulla base di quelle preparate da T. Calamoneri e G. Bongiovanni per il corso di Informatica Generale tenuto a distanza nell'A.A. 2019/20

Introduzione alla notazione asintotica (1)

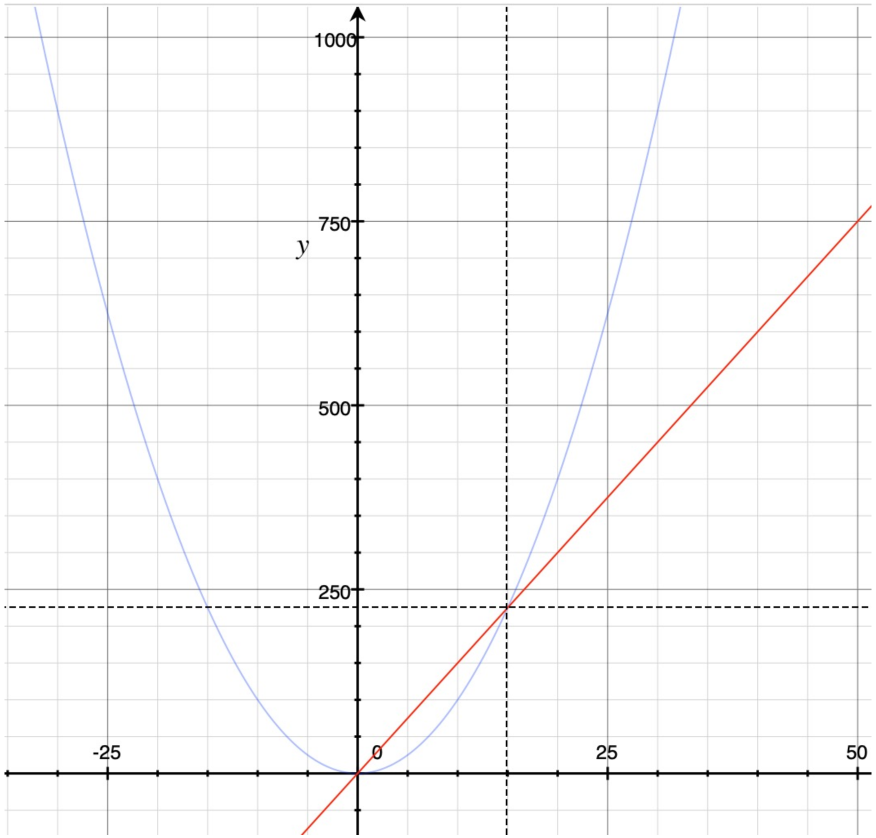
- Vogliamo valutare l'efficienza di un algoritmo, così da poterlo confrontare con algoritmi diversi che risolvono lo stesso problema.
- Lo facciamo in termini di ***costo computazionale***, ovvero del **tempo di esecuzione** di un algoritmo e delle sue **necessità in termini di memoria**.
- Prediligiamo il **tempo di esecuzione** all'occupazione di memoria.

Introduzione alla notazione asintotica (2)

In matematica la **notazione asintotica** permette di confrontare il tasso di crescita (comportamento asintotico) di una funzione nei confronti di un'altra.

$f(n)=15n+1$

$g(n)=n^2$



Introduzione alla notazione asintotica (3)

In informatica, il calcolo asintotico è utilizzato per analizzare la complessità di un algoritmo. In particolar modo, per stimare quanto aumenta il tempo al crescere della dimensione *n* dell'input.

Notazione asintotica *O*:
La notazione *O grande* è il limite superiore asintotico
Notazione asintotica *Ω*:
La notazione *Omega* è il limite inferiore asintotico
Notazione asintotica *Θ*:
La notazione *Teta* è il limite asintotico stretto

Introduzione alla notazione asintotica (4)

Tale valutazione ha senso quando la **dimensione dell'input è sufficientemente grande**. Per questo si parla di ***efficienza asintotica degli algoritmi***.

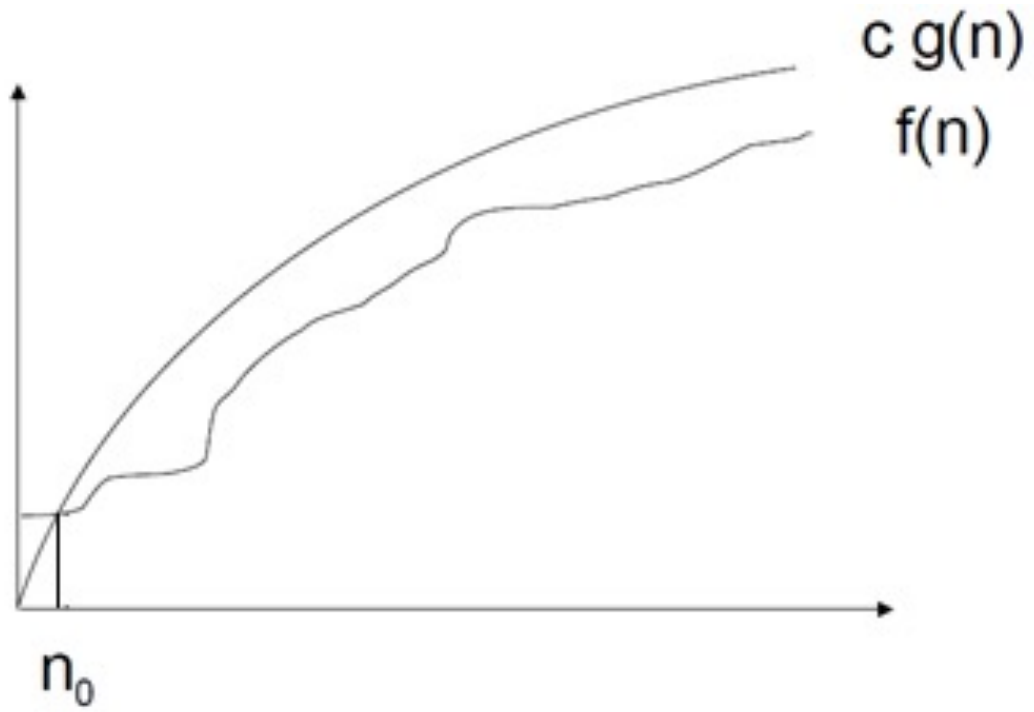
Notazione *O* – O grande (1)

Date due funzioni $f(n)$, $g(n) \geq 0$ si dice che

$$f(n) \text{ è in } O(g(n))$$

se esistono due costanti c ed n_0 tali che

$$0 \leq f(n) \leq c\, g(n) \text{ per ogni } n \geq n_0$$



In $O(g(n))$ troviamo tutte le funzioni che risultano «dominate» dalla funzione $g(n)$

Notazione *O* – O grande (2)

Esempio. $\boldsymbol{f(n) = 3n + 3}$

$f(n)$ è in $\boldsymbol{O(n^2)}$ in quanto, posto $c = 6$:

$$\boldsymbol{cn^2 \geq 3n + 3} \text{ per ogni } n \geq 1.$$

Ma $f(n)$ è anche in $\boldsymbol{O(n)}$ in quanto:

$$\boldsymbol{cn \geq 3n + 3} \text{ per ogni } n \geq 1 \text{ se } c \geq 6,$$

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

💡 data $f(n)$, esistono ***infinite funzioni*** $g(n)$ per cui $f(n)$ risulta in $O(g(n))$.

Notazione *O* – O grande (3)

Esempio. Sia $f(n) = n^2 + 4n$

$f(n)$ è in $O(n^2)$ in quanto:

$cn^2 \geq n^2 + 4n$ per ogni n se $c \geq 5$

oppure per ogni $n \geq 4/(c-1)$ se $c > 1$.

Notazione *O* – O grande (4)

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

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

Dimostriamo, per induzione su *m*, che ***f(n)* è in *O(n^m)***

Caso base:

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

Notazione *O* – O grande (5)

Esempio (segue)

Ipotesi induttiva:

$\sum_{i=0}^{k} a_i n^i$ è un $O(n^k)$ per ogni $k<m$

cioè esiste una costante c' tale che $\sum_{i=0}^{k} a_i n^i \leq c'\, n^k$.

Passo induttivo:

Dobbiamo dimostrare che

$f(n) = \sum_{i=0}^{m} a_i n^i$ è in $O(n^m)$

cioè che esiste una costante c tale che:

$$\sum_{i=0}^{m} a_i n^i \leq c\, n^m.$$

Notazione *O* – O grande (6)

Esempio (segue)

Si osservi che $f(n)$ si può scrivere come:

$$f(n) = \sum_{i=0}^{m} a_i n^i = a_m n^m + \sum_{i=0}^{k} a_i n^i = a_m n^m + h(n)$$

con $k<m$ e che, per l'ipotesi induttiva:

$$h(n) \leq c' n^k$$

Ora:

$$f(n) = a_m n^m + h(n) \leq a_m n^m + c' n^k \leq a_m n^m + c' n^m + = (c' + a_m) n^m.$$

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

Notazione *O* – O grande (7)

Esempio. Sia ***f(n) = log n.*** *f(n)* è in $\boldsymbol{O(\sqrt{n})}$.

Più in generale, $log^a\ n = O(n^{1/b})$ per ogni a,b ≥ *1*

Cioè:
un poli-logaritmo è dominato da una qualunque radice

Notazione *O* – O grande (8)

Esempio. Sia $\boldsymbol{f(n) = n^{1/a}}$. $f(n)$ è in $\boldsymbol{O(n)}$ per ogni a ≥ 2

Più in generale, $n^{1/a} = O(n^b)$ per ogni a,b ≥ 1

Cioè:
Una radice è dominata da un qualunque polinomio

Notazione *O* – O grande (9)

Esempio. Sia $\boldsymbol{f(n) = n^a}$. $f(n)$ è in $\boldsymbol{O(2^n)}$ per ogni a ≥ 1

Più in generale, $n^a = O(b^n)$ per ogni a ≥ 1, e b ≥ 2

Cioè:
Un polinomio è dominato da un qualunque esponenziale

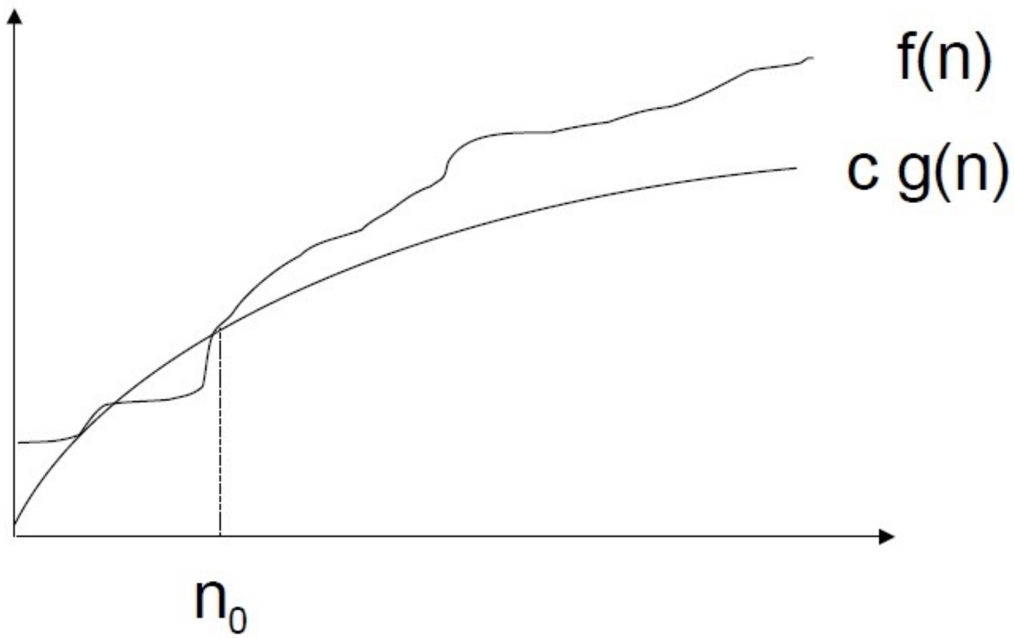
Notazione Ω - Omega (1)

Date due funzioni $f(n)$, $g(n) \geq 0$ si dice che

$$f(n) \text{ è in } \Omega(g(n))$$

se esistono due costanti c ed n_0 tali che

$$f(n) \geq c\, g(n) \text{ per ogni } n \geq n_0$$



In $\Omega(g(n))$ troviamo tutte le funzioni che «dominano» la funzione $g(n)$

Notazione Ω - Omega (2)

Esempio. Sia $\boldsymbol{f(n) = 2n^2 + 3}$

$f(n)$ è in $\boldsymbol{\Omega(n)}$ in quanto

$$\boldsymbol{2n^2 + 3 \geq cn} \text{ per qualunque } n \text{ se } c = 1$$

Ma $f(n)$ è anche in $\boldsymbol{\Omega(n^2)}$ in quanto

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

Notazione Ω - Omega (3)

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

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

La dimostrazione che *f(n)* è in $\Omega(n^m)$ è analoga alla dimostrazione che *f(n)* è in $O(n^m)$ e perciò viene lasciata come esercizio.

Notazione Ω - Omega (4)

Esempio. Sia $\boldsymbol{f(n) = \sqrt{n}}$. $f(n)$ è in $\boldsymbol{\Omega(\log n)}$.

Più in generale, $n^{1/b} = \Omega(\log^a n)$ per ogni a,b ≥ 1

Cioè:
una radice domina qualunque poli-logaritmo

Notazione Ω - Omega (5)

Esempio. Sia ***f(n)* = $\mathbf{2^n}$.** *f(n)* è in **Ω*(n^a)*** per ogni a ≥ *1*

Più in generale, $b^n = \Omega(n^a)$ per ogni a ≥ *1*, e b ≥ *2*

Cioè:
Un esponenziale domina un qualunque polinomio

Notazione O e Ω - considerazioni (1)

Abbiamo visto che in entrambe le notazioni *O* e Ω, per ogni funzione *f(n)* sia possibile trovare più funzioni *g(n)*.

In effetti *O(g(n))* e *Ω(g(n))* sono ***insiemi di funzioni***, e dire "*f(n)* è in *O(g(n)*" oppure "*f(n)* = *O(g(n)*" ha il significato di "*f(n)* appartiene a *O(g(n))*".

Notazione O e Ω - considerazioni (2)

Tuttavia, 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**. La definizione che segue formalizza questo concetto intuitivo.

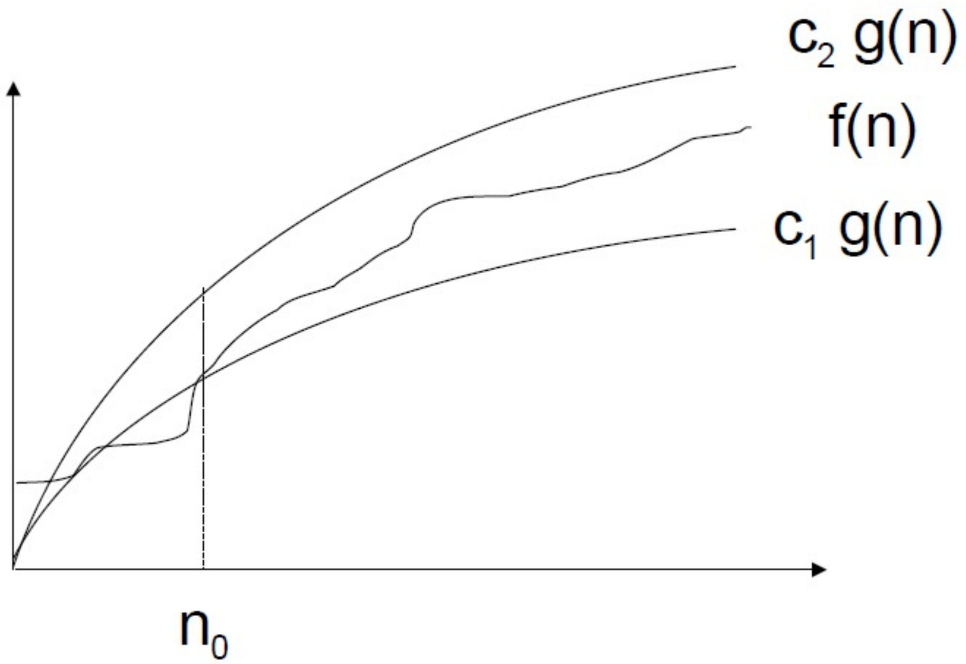
Notazione Θ – Teta (1)

Date due funzioni $f(n)$, $g(n) \geq 0$ si dice che

$f(n)$ è in $\Theta(g(n))$

se esistono tre costanti c_1, c_2 ed n_0 tali che

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



Notazione Θ – Teta (2)

Esempio. Sia $\boldsymbol{f(n) = 3n + 3}$

f(n) è in $\boldsymbol{\Theta(n)}$ ponendo, ad esempio:

$$c_1 = 3,\ c_2 = 4,\ n_0 = 3$$

Infatti:

$$3n \leq 3n + 3 \leq 4n \text{ per } n \geq 3$$

Notazione Θ – Teta (3)

Esempio. Dimostrare che $f(n) = \log_a n = \Theta(\log_b n)$ per ogni a,b>0.

Basta usare la formula per il cambio di base dei logaritmi:

$$\log_a n = \log_b n \log_a b = c \log_b n$$

Il cambio di base è dunque asintoticamente irrilevante e per questo nella notazione asintotica la base del logaritmo viene spesso omessa.

Notazione *Θ* – Teta (4)

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

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

La dimostrazione che *f(n)* è in $\Theta(n^m)$ discende dall'aver dimostrato che

$$\sum_{i=0}^{m} a_i n^i \text{ è sia in } O(n^m) \text{ che in } \Omega(n^m)$$

Calcolo della notaz. asint. tramite limiti

- se $\lim_{n\to\infty} \frac{f(n)}{g(n)} = k > 0$ allora $f(n)=\Theta(g(n))$;
- se $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$ allora $f(n)=\Omega(g(n))$ ma $f(n)\neq\Theta(g(n))$;
- se $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$ allora $f(n)=O(g(n))$ ma $f(n)\neq\Theta(g(n))$.

Ovviamente, quando il limite non esiste, questo metodo non si può usare e bisogna procedere diversamente.

Algebra della notazione asintotica

Per semplificare il calcolo del costo computazionale asintotico degli algoritmi si possono sfruttare delle semplici regole che dapprima enunciamo, chiarendole con degli esempi, ed in un secondo momento dimostriamo.

Regole sulle costanti moltiplicative

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

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

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

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 in $O(g(n))$ e $d(n)$ è in $O(h(n))$

allora $\boldsymbol{f(n)+d(n)}$ è in $\boldsymbol{O(g(n)+h(n))} = \boldsymbol{O(max(g(n),h(n)))}$.

2B: Per ogni $f(n), d(n) > 0$,

se $f(n)$ è in $\Omega(g(n))$ e $d(n)$ è in $\Omega(h(n))$

allora $\boldsymbol{f(n)+d(n)}$ è in $\boldsymbol{\Omega(g(n)+h(n))} = \boldsymbol{\Omega(max(g(n),h(n)))}$.

2C: Per ogni $f(n), d(n) > 0$,

se $f(n)$ è in $\Theta(g(n))$ e $d(n)$ è in $\Theta(h(n))$

allora $\boldsymbol{f(n)+d(n)}$ è in $\boldsymbol{\Theta(g(n)+h(n))} = \boldsymbol{\Theta(max(g(n),h(n)))}$.

Informalmente:

😀 **le notazioni asintotiche commutano con l'operazione di somma**.

Regole sulla commutatività col prodotto

3A: Per ogni $f(n), d(n) > 0$,
se $f(n)$ è in $O(g(n))$ e $d(n)$ è in $O(h(n))$
allora $\boldsymbol{f(n)d(n)}$ è in $\boldsymbol{O(g(n)h(n))}$.

3B: Per ogni $f(n), d(n) > 0$,
se $f(n)$ è in $\Omega(g(n))$ e $d(n)$ è in $\Omega(h(n))$
allora $\boldsymbol{f(n)d(n)}$ è in $\boldsymbol{\Omega(g(n)h(n))}$.

3C: Per ogni $f(n), d(n) > 0$,
se $f(n)$ è in $\Theta(g(n))$ e $d(n)$ è in $\Theta(h(n))$
allora $\boldsymbol{f(n)d(n)}$ è in $\boldsymbol{\Theta(g(n)h(n))}$.

Informalmente:
😀 **le notazioni asintotiche commutano con l'operazione di prodotto.**

Esempi di applicazione delle regole (1)

Esempio 1

Trovare il limite asintotico stretto per $f(n) = 3n^2 + 7$.

$$3n^2 = O(n^2) \text{ e } 7 = O(n^2) \text{ quindi } 3n^2 + 7 = O(n^2).$$

Esempio 2

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 3

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

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

Esempi di applicazione delle regole (2)

Esempio 4

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

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

le *costanti moltiplicative si possono ignorare solo se* non *sono all'esponente*.

Esempio 5

Trovare il limite asintotico stretto per $f(n) = \log^n n + 8 \cdot 2^{n \log n} + 3$

$$\log^n n + 8 \cdot 2^{n \log n} + 3 = \log^n n + 8 n^n + 3 =$$
$$= \Theta(\log^n n) + \Theta(1)\, \Theta(n^n) + \Theta(1) = \Theta(n^n).$$

Dimostrazione della regola 1A

Regola:

Per ogni $k > 0$ e per ogni $f(n) \geq 0$,

se $f(n)$ è in $O(g(n))$

allora anche $k\ f(n)$ è in $O(g(n))$.

Dimostrazione

Per ipotesi, $f(n)$ è in $O(g(n))$ 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 in $O(g(n))$**.

CVD

Dimostrazione della regola 2A

Regola:

Per ogni $f(n)$, $d(n) > 0$,

se $f(n)$ è in $O(g(n))$ e $d(n)$ è in $O(h(n))$

allora $f(n)+d(n)$ è in $O(g(n)+h(n)) = O(max(g(n),h(n)))$.

Dimostrazione

Se $f(n)$ è in $O(g(n))$ e $d(n)$ è in $O(h(n))$ 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 $n \geq max(n'_0, n''_0)$

Da ciò segue che **$f(n) + d(n)$ è in $O(g(n)+h(n))$**.

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 in $O(max(g(n), h(n)))$**. CVD

Dimostrazione della regola 3A

Regola:

Per ogni $f(n)$, $d(n) > 0$,

se $f(n)$ è in $O(g(n))$ e $d(n)$ è in $O(h(n))$
allora $f(n)d(n)$ è in $O(g(n)h(n))$.

Dimostrazione

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

$$f(n) \le c'g(n) \text{ per ogni } n \ge n'_0 \text{ e } d(n) \le c''h(n) \text{ per ogni } n \ge n''_0$$

Allora:

$$f(n)d(n) \le c'c''g(n)h(n) \text{ per ogni } n \ge max(n'_0, n''_0)$$

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

CVD

Le dimostrazioni delle altre regole, che coinvolgono le notazioni Ω e Θ, sono lasciate per esercizio.

Esercizi per casa

Esercizi per casa (1)

- Dimostrare tutte le regole sull'algebra della notazione asintotoca.
- Calcolare l'andamento asintotico delle seguenti funzioni:
  - $f(n) = n^2 \log n$
  - $f(n) = 3n \log n + 2n^2$
  - $f(n) = 2^{\log n/2} + 5n$
  - $f(n) = 4^{\log n}$
  - $f(n) = (\sqrt{2})^{\log n}$

Esercizi per casa (2)

Classificare le seguenti funzioni per ordine di crescita, vale a dire, trovare un ordinamento $g_1, g_2, \dots, g_n$ delle funzioni che soddisfi:

$$g_1 = \Omega(g_2),\ g_2 = \Omega(g_3),\ \ldots$$

Partiziona poi la lista in classi di equivalenza in modo che le funzioni $f(n)$ e $g(n)$ sono nella stessa classe se e solo se $f(n)=\Theta(g(n))$.

$(\sqrt{2})^{\log n}$ $\quad n^2$ $\quad n!$ $\quad (\log n)!$ $\quad \log n^n$ $\quad n^3$

$\log (n!)$ $\quad 2^n$ $\quad n \cdot 2^n$ $\quad \log n$ $\quad n \log n$

$(3/2)^n$ $\quad e^n$ $\quad 5$ $\quad 4^{\log n}$ $\quad 2/3$ $\quad \log^2 n$