Tempi di Esecuzione

I tempi di esecuzione degli algoritmi sono rappresentati da funzioni matematiche su cui si analizza il **comportamento asintotico** ($n \to \infty$)

$$f : \mathbb{N} \to \mathbb{N}$$

Esempio

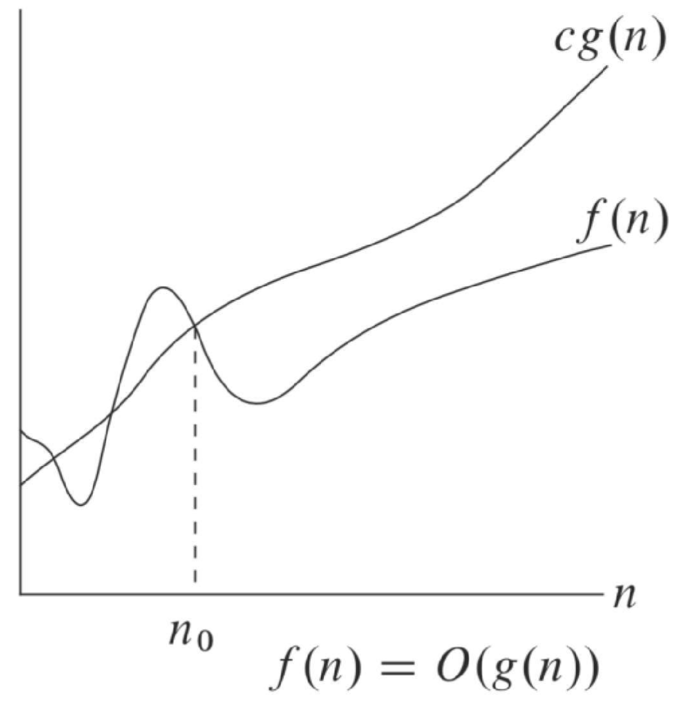
$$n \log n, n^2 + 3n + 2, n!, \lceil \sqrt{n+2} \rceil$$

Ordini di Grandezza

Possiamo distinguere **3 principali ordini di grandezza**

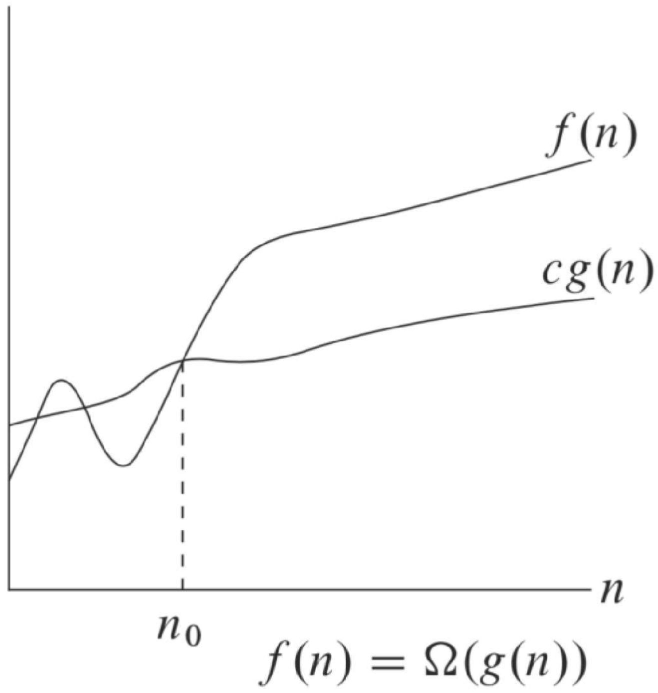
Definizione: O grande

$$f(n) = O(g(n)) \iff \exists\, n_0, c > 0 \mid \forall n > n_0, f(n) \le c \cdot g(n)$$



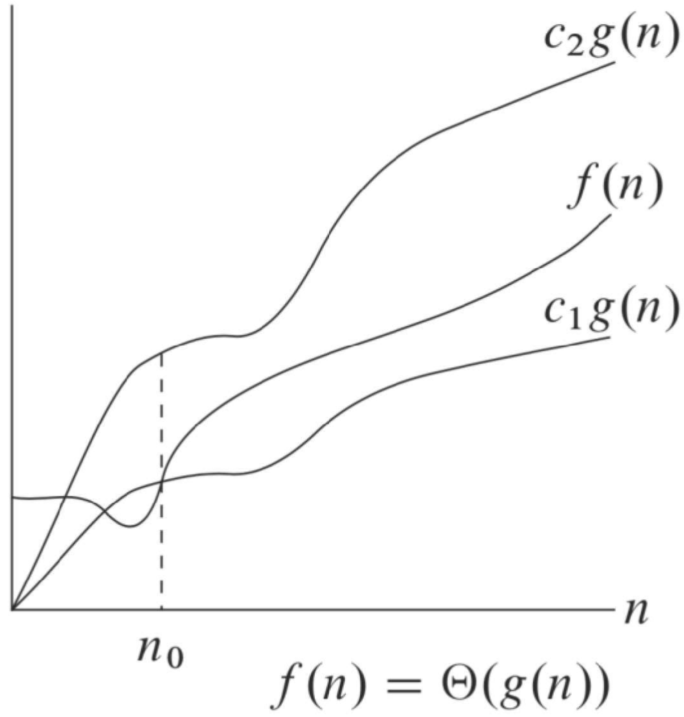
Definizione: Ω grande

$$f(n) = \Omega(g(n)) \iff \exists\, n_0, c > 0 \mid \forall n > n_0, f(n) \ge c \cdot g(n)$$



Definizione: Θ grande

$$f(n) = \Theta(g(n)) \iff f(n) = O(g(n)) \land f(n) = \Omega(g(n))$$



O grande e Omega grande presentano anche le alternative piccole

Definizione: $\circ$ e ω piccoli

$$f(n) = \circ(g(n)) \iff f(n) = O(g(n)) \land f(n) \neq \Theta(g(n))$$
$$f(n) = \omega(g(n)) \iff f(n) = \Omega(g(n)) \land f(n) \neq \Theta(g(n))$$

Proprietà

Proprietà: Transitività

$$
\begin{aligned}
f(n) = \Theta(g(n)), g(n) = \Theta(h(n)) &\implies f(n) = \Theta(h(n)) \\
f(n) = O(g(n)), g(n) = O(h(n)) &\implies f(n) = O(h(n)) \\
f(n) = \Omega(g(n)), g(n) = \Omega(h(n)) &\implies f(n) = \Omega(h(n)) \\
f(n) = \circ(g(n)), g(n) = \circ(h(n)) &\implies f(n) = \circ(h(n)) \\
f(n) = \omega(g(n)), g(n) = \omega(h(n)) &\implies f(n) = \omega(h(n))
\end{aligned}
$$

Proprietà: Riflessività

$$
\begin{aligned}
f(n) = \Theta(f(n)) \\
f(n) = O(f(n)) \\
f(n) = \Omega(f(n))
\end{aligned}
$$

Proprietà: Simmetria e Simmetria trasposta

$$
\begin{aligned}
f(n) = \Theta(g(n)) &\iff g(n) = \Theta(f(n)) \\
f(n) = O(g(n)) &\iff g(n) = \Omega(f(n)) \\
f(n) = \circ(g(n)) &\iff g(n) = \omega(f(n))
\end{aligned}
$$

⚠ Relazione d'ordine parziale

Esistono f e g **non confrontabili**, spesso quando sono presenti **funzioni oscillanti**

$$f(n) = n, g(n) = n^{1+\sin(n)}$$

Calcolo degli Ordini

Calcolo con Limiti

Possiamo notare tre uguaglianze utili per il **calcolo degli ordini di grandezza**

$$
\begin{aligned}
\lim_{n\to\infty} \frac{f(n)}{g(n)} = c \neq 0 &\implies f(n) = \Theta(g(n)) \\
\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 &\implies f(n) = \circ(g(n)) \\
\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty &\implies f(n) = \omega(g(n))
\end{aligned}
$$

Calcolo con Logaritmi

Per semplificare il calcolo quando sono **presenti esponenziali**, ricordiamo

$$
\begin{aligned}
\log(f(n)) = \circ(\log(g(n))) &\implies f(n) = \circ(g(n)) \\
\log(f(n)) = \omega(\log(g(n))) &\implies f(n) = \omega(g(n)) \\
\log(f(n)) = \Theta(\log(g(n))) &\not\Longrightarrow f(n) = \Theta(g(n))
\end{aligned}
$$

Proprietà: Taglia di n

Sia $n > 1$ un numero in base b

$$\log_b n = |n| \pm 1$$

Dove $|n|$ è la taglia di n in base b

Proprietà: Omissione della Base

Siano a, b due basi diverse

$$\log_a(n) = \log_a(b) \log_b(n)$$

Questo ci permette di **omettere la base dei logaritmi** nelle formule in quanto rimangono dello **stesso ordine di grandezza**

Proprietà: Polilogaritmi e Polinomi

Un **polilogaritmo** (logaritmi moltiplicati insieme) sarà sempre o piccolo di un **polinomio**

$$\text{polilogaritmo} = \circ(\text{ polinomio })$$

In conclusione, otteniamo la seguente **tassonomia**:

Tassonomia

Sia $M > 0$ **grande a piacere**, $\epsilon > 0$ **piccola a piacere**, $a > 1$

$$\begin{aligned} \log(n)^M &= \circ(n^\epsilon) \\ n^M &= \circ(a^{\epsilon n}) \\ a^{Mn} &= \circ(\epsilon n!) \\ Mn! &= \circ(n^{\epsilon n}) \end{aligned}$$