Notazione Asintotica

Sommario:

- Ordini di grandezza: la notazione asintotica
- La velocità di crescita delle funzioni

Notazione Asintotica

- La notazione asintotica è un modo per indicare certi insiemi di funzioni caratterizzati da specifici comportamenti all'infinito
- Questi insiemi sono indicati dalle lettere

$$\Theta \; O \; \Omega \; o \; \omega$$

- Quando una funzione f(n) appartiene ad uno di questi insiemi lo si indica equivalentemente come
  - $f(n) \in \Theta(n^2)$
  - $f(n) = \Theta(n^2)$
- La seconda notazione è inusuale ma vedremo che ha dei vantaggi di uso

Notazione $\Theta(g(n))$

- Con la notazione $\Theta(g(n))$ si indica l'insieme di funzioni $f(n)$ che soddisfano la seguente condizione

  $\Theta(g(n))=\{f(n): \exists\ c_1, c_2, n_0$ tali che

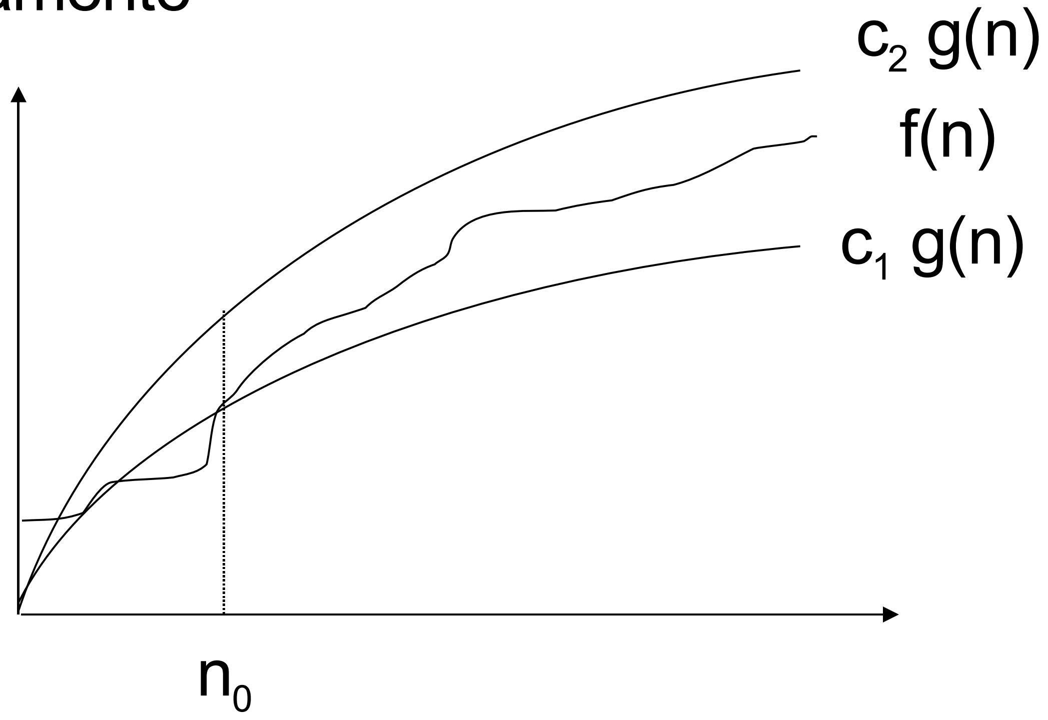
  $\forall\ n \geq n_0$

  $0 \leq c_1\, g(n) \leq f(n) \leq c_2\, g(n)\ \}$

- Ovvero $f(n)$ appartiene a $\Theta(g(n))$ se esistono due costanti c_1, c_2 tali che essa possa essere schiacciata fra $c_1\, g(n)$ e $c_2\, g(n)$ per n sufficientemente grandi

Notazione $\Theta(g(n))$

- Graficamente



Notazione O(g(n))

- Con la notazione O(g(n)) si indica l'insieme di funzioni f(n) che soddisfano la seguente condizione

  $O(g(n))=\{f(n): \exists\, c, n_0$ tali che

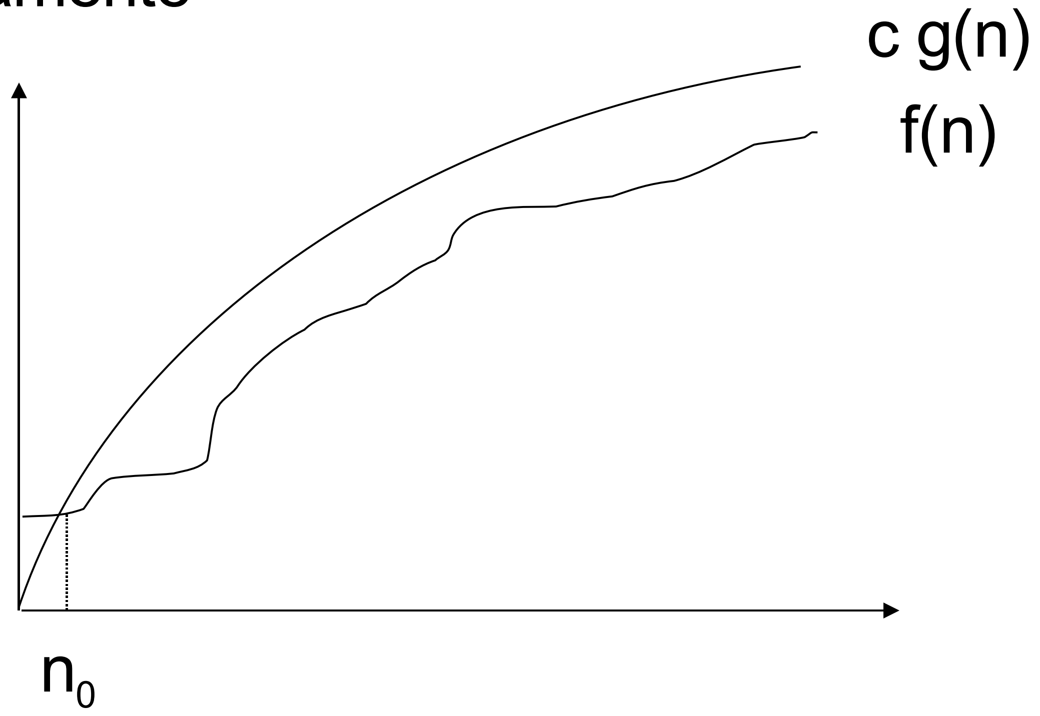
  $\forall\, n \geq n_0$

  $0 \leq f(n) \leq c\, g(n)\}$

- Ovvero f(n) appartiene a O(g(n)) se esiste una costante c tali che essa possa essere maggiorata da c g(n) per n sufficientemente grandi

Notazione O(g(n))

- Graficamente



Notazione $\Omega(g(n))$

- Con la notazione $\Omega(g(n))$ si indica l'insieme di funzioni f(n) che soddisfano la seguente condizione

  $\Omega(g(n))=\{f(n): \exists\ c, n_0$ tali che

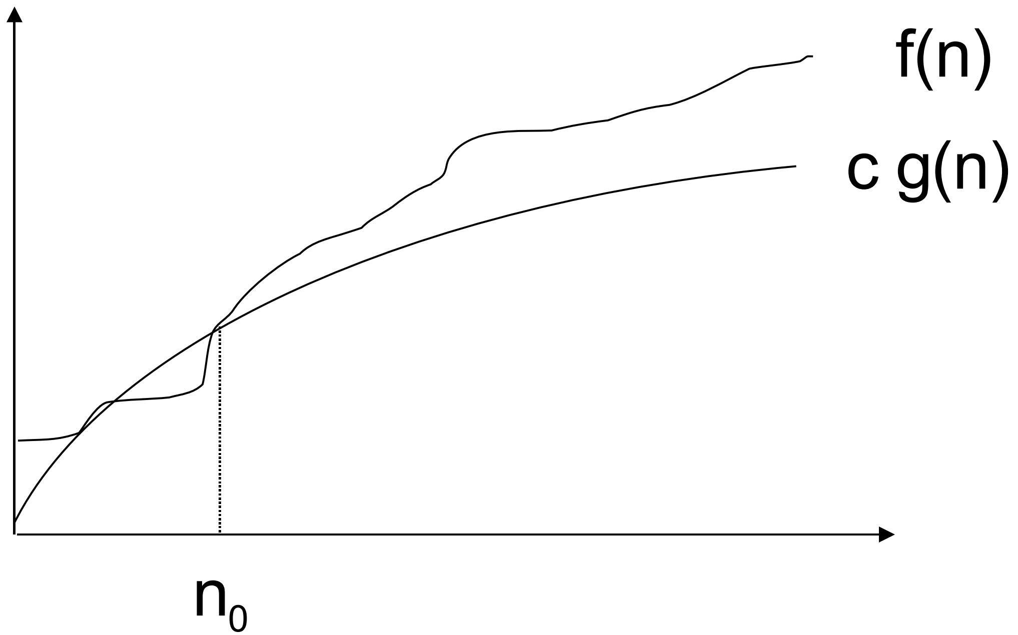
  $\forall\ n \geq n_0$

  $0 \leq c\ g(n) \leq f(n)\ \}$

- Ovvero f(n) appartiene a $\Omega(g(n))$ se esiste una costante c tali che essa sia sempre maggiore di c.g(n) per n sufficientemente grandi

Notazione $\Omega(g(n))$

- Graficamente



Notazione o(g(n))

- Il limite asintotico superiore può essere stretto o no
- $2\ n^2 = O(n^2)$ è stretto
- $2\ n = O(n^2)$ non è stretto
- Con la notazione o(g(n)) si indica un limite superiore non stretto
- Formalmente, con la notazione o(g(n)) si indica l'insieme di funzioni f(n) che soddisfano la seguente condizione

  $o(g(n))=\{f(n): \forall\ c>0\ \exists\ n_0$ tali che

  $\forall\ n \geq n_0$

  $0 \leq\ f(n)\ \leq\ c\ g(n)\ \}$

Notazione o(g(n))

- La definizione di o() differisce da quella di O() per il fatto che la maggiorazione in o() vale per qualsiasi costante positiva mentre in O() vale per una qualche costante
- L'idea intuitiva è che la f(n) diventa trascurabile rispetto alla g(n) all'infinito ovvero

  $\lim_{n \to \infty} f(n)/g(n)=0$

Notazione $\omega(g(n))$

- Analogamente nel caso di limite inferiore non stretto si definisce che con la notazione $\omega(g(n))$ si indica l'insieme di funzioni f(n) che soddisfano la seguente condizione

  $\omega(g(n))=\{f(n): \forall\ c>0\ \exists\ n_0$ tali che

  $\forall\ n \geq n_0$

  $0 \leq c\ g(n) \leq f(n)\}$

- Qui l'idea intuitiva è che sia la g(n) a diventare trascurabile rispetto alla f(n) all'infinito ovvero

  $\lim_{n\to\infty} f(n)/g(n)=\infty$

Tralasciare i termini di ordine più basso

- Giustifichiamo perché è possibile tralasciare i termini di ordine più basso, ovvero perché possiamo scrivere $1/2\ n^2 - 3\ n = \Theta(n^2)$
- Dalla definizione di $\Theta(g(n))$ si ha che si devono trovare delle costanti c_1, c_2 tali che $1/2\ n^2 - 3\ n$ possa essere schiacciata fra $c_1\ n^2$ e $c_2\ n^2$ per n sufficientemente grandi, ovvero per $n>n_0$

  $c_1\ n^2 \leq 1/2\ n^2 - 3\ n \leq\ c_2\ n^2$

  $c_1\ n^2 \leq 1/2\ n^2 - 3\ n$ è vera per $n \geq 7$ e per $c_1 \geq 1/14$

  $1/2\ n^2 - 3\ n \leq\ c_2\ n^2$ è vera per $n \geq 1$ e per $c_2 \geq 1/2$
- Quindi per $n_0=7$ $c_1 = 1/14$ e $c_2 = 1/2$ si è soddisfatta la tesi (altri valori sono possibili ma basta trovarne alcuni)

Tralasciare i termini di ordine più basso

- Intuitivamente si possono tralasciare i termini di ordine più basso perché una qualsiasi frazione del termine più alto prima o poi sarà più grande di questi
- Quindi, nel caso dei polinomi, assegnando a c_1 un valore più piccolo del coefficiente del termine più grande e a c_2 un valore più grande dello stesso consente di soddisfare le disegualianze della definizione di $\Theta(g(n))$
- Il coefficiente del termine più grande può poi essere ignorato perché cambia solo i valori delle costanti

Nota

- In sintesi si può sempre scrivere che

  $a\ n^2 + b\ n + c = \Theta(n^2)$
- ovvero

  $\Sigma_{j=o..d}\ a_j n^j = \Theta(n^d)$

- inoltre dato che una costante è un polinomio di grado 0 si scrive:

  $c = \Theta(n^0) = \Theta(1)$

Uso della notazione asintotica

- Dato che il caso migliore costituisce un limite inferiore al tempo di calcolo, si usa la notazione $\Omega(g(n))$ per descrivere il comportamento del caso migliore
- Analogamente dato che il caso peggiore costituisce un limite superiore al tempo di calcolo, si usa la notazione $O(g(n))$ per descrivere il comportamento del caso peggiore
- Per l'algoritmo di insertion sort abbiamo trovato che nel caso migliore si ha $T(n)= \Omega(n)$ e nel caso peggiore $T(n)=O(n^2)$

La notazione asintotica nelle equazioni

- Seguendo la notazione $n = O(n)$ possiamo pensare di scrivere anche espressioni del tipo

$$2n^2+3n+1= 2n^2+O(n)$$

- Il significato di questa notazione è che con $O(n)$ vogliamo indicare una anonima funzione che non ci interessa specificare (ci basta che sia limitata superiormente da n)
- Nel nostro caso questa funzione è proprio $3n+1$ che è $O(n)$
- Tramite l'uso della notazione asintotica possiamo eliminare da una equazione dettagli inessenziali

La notazione asintotica nelle equazioni

- La notazione asintotica può anche apparire a sinistra di una equazione come in

$$2n^2+O(n)= O(n^2)$$

- Il significato è che indipendentemente da come viene scelta la funzione anonima a sinistra è sempre possibile trovare una funzione anonima a destra che soddisfa l’equazione per ogni n
- In questo modo possiamo scrivere:

$$2n^2+3n+1= 2n^2+O(n)=O(n^2)$$

Le funzioni di interesse

- O(1) il tempo costante è caratteristico di istruzioni che sono eseguite una o al più poche volte.

- O(log n) il tempo logaritmico è caratteristico di programmi che risolvono un problema di grosse dimensioni riducendone la dimensione di un fattore costante e risolvendo i singoli problemi più piccoli. quando il tempo di esecuzione è logaritmico il programma rallenta solo leggermente al crescere di *n*: se *n* raddoppia *log n* cresce di un fattore costante piccolo.

Le funzioni di interesse

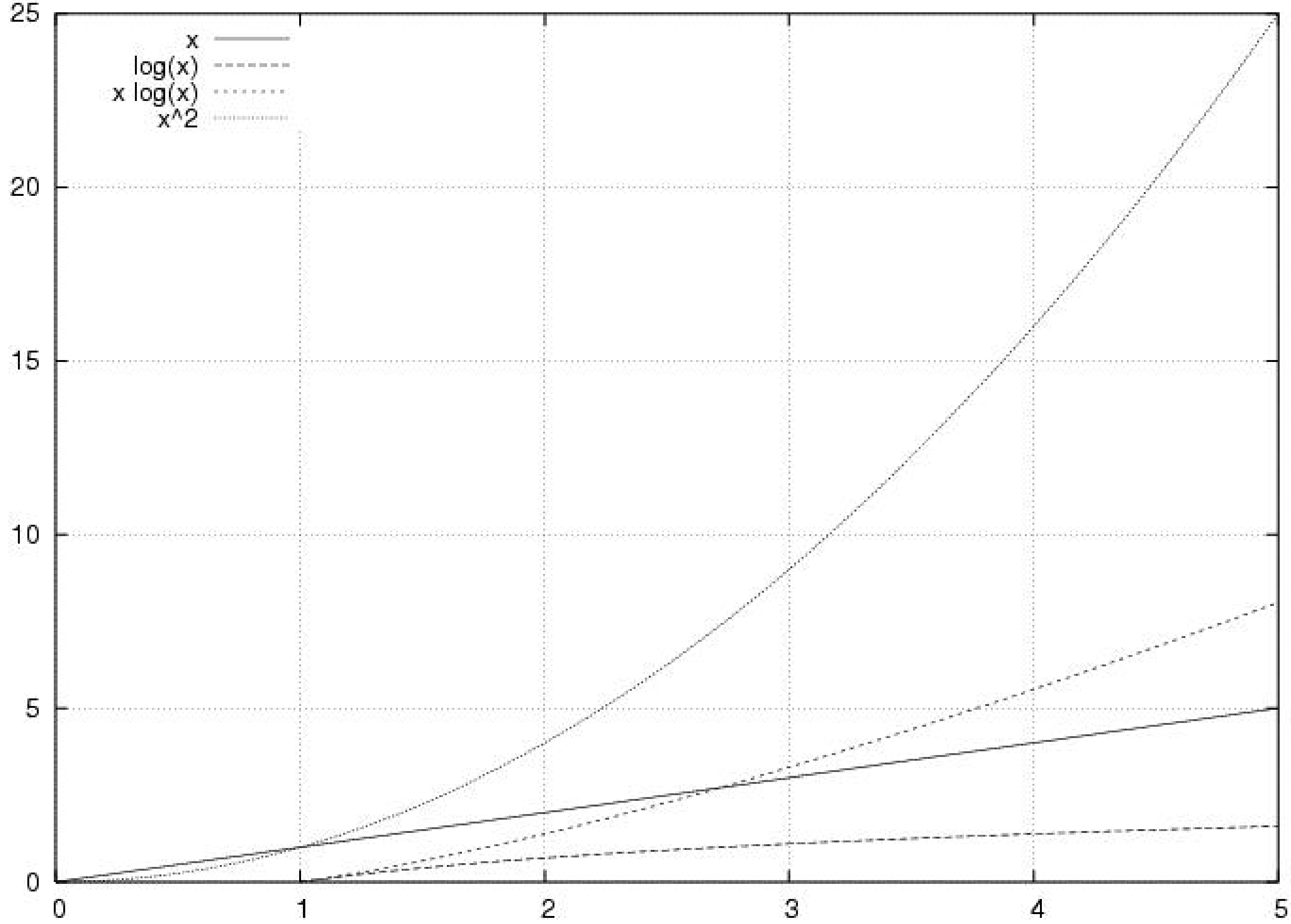
- O(n) il tempo lineare è caratteristico di programmi che eseguono poche operazioni su ogni elemento dell'input. Se la dimensione dell'ingresso raddoppia, raddoppia anche il tempo di esecuzione.

- O(n log n) il tempo n log n è caratteristico di programmi che risolvono un problema di grosse dimensioni riducendoli in problemi più piccoli, risolvendo i singoli problemi più piccoli e ricombinando i risultati per ottenere la soluzione generale. Se *n* raddoppia *n log n* diventa poco più del doppio.

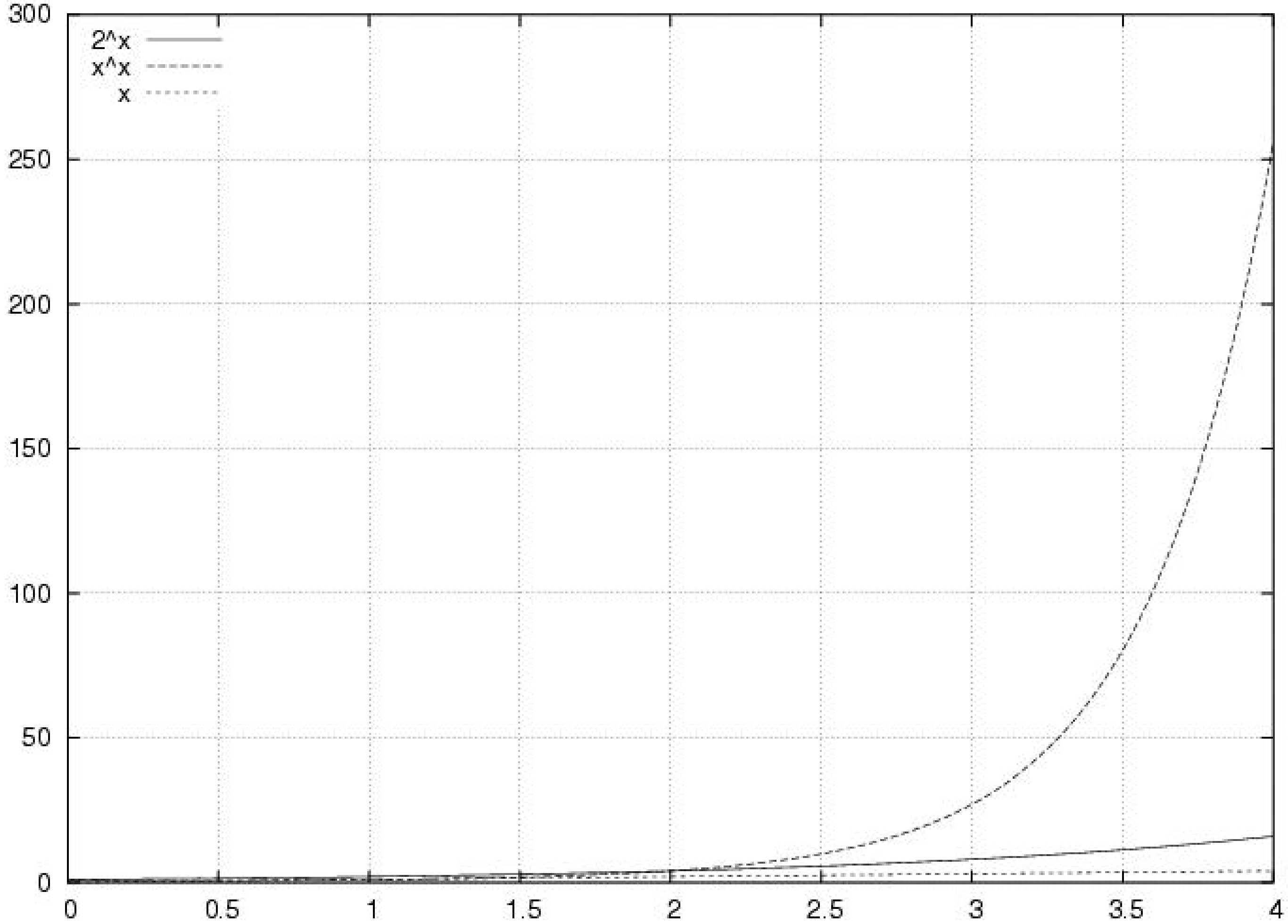
Le funzioni di interesse

- $O(n^2)$ il tempo quadratico è caratteristico di programmi che elaborano l'input a coppie. Algoritmi con tempo quadratico si usano per risolvere problemi abbastanza piccoli. Se n raddoppia n^2 quadruplica.
- $O(2^n)$ il tempo esponenziale è caratteristico di programmi che elaborano l'input considerando tutte le possibili permutazioni. Rappresentano spesso la soluzione naturale più diretta e facile di un problema. Algoritmi con tempo esponenziale raramente sono applicabili a problemi pratici. Se l'input raddoppia il tempo di esecuzione viene elevato al quadrato

Crescita delle funzioni



Crescita delle funzioni



La conversione dei secondi

- Secondi

10^2	1.7 minuti
10^4	2.8 ore
10^5	1.1 giorni
10^6	1.6 settimane
10^7	3.8 mesi
10^8	3.1 anni
10^9	3.1 decenni
10^{10}	3.1 secoli
10^{11}	mai

Andamento dei tempi di calcolo

N	log N	N log N	N^2	2^N
10	3	30	10^2	10^3
10^2	7	7 10^2	10^4	10^30
10^3	10	10 ^4	10^6	10^300
10^6	17	2 10^7	10^12	-
10^12	32	3 10^13	10^24	-

Andamento dei tempi di calcolo

N	N	log N	N log N	N^2	2^N
10	istantaneo	istantaneo	istantaneo	istantaneo	secondi
10^2	istantaneo	istantaneo	istantaneo	istantaneo	mai
10^3	istantaneo	istantaneo	istantaneo	secondi	mai
10^6	secondi	istantaneo	secondi	settimane	-
10^12	settimane	istantaneo	mesi	mai	-

Tempo impiegato da un calcolatore capace di 10^6 operazioni al secondo