Titolo nota 02/03/2016

$T(n)$

MERGE_SORT (A, p, r)

Divide et Impera

- Dividi
- Risolvi ricorsivamente / direttamente
- Combina

if $p < r$

dividi → $q = \frac{(p+r)}{2}$ $\Theta(1)$

risolvi → MERGE_SORT (A, p, q) $T(n/2)$

MERGE_SORT (A, q+1, r) $T(n/2)$

combina → MERGE (A, p, q, r) → $\Theta(n)$

corretto

Induzione

base: $n = 1$ $(p = r)$ vero

passo induttivo: $n/2 \Rightarrow n$ vero

Titolo nota 02/03/2016

Analisi Merge Sort

- Correttezza
- Complessità tempo
- " spazio $\Theta(n)$

Merge

invariante: $A[p..k-1]$ contiene i primi $k-p$ elementi ordinati e $L[i]$ e $R[j]$ contengono i 2 minimi degli elementi da ordinare

inizio $k=1$: $A[p..k-1] = \emptyset$, $L[1]$ e $R[1]$ contengono i 2 minimi

conservazione: k: $A[p..k-1, k] \rightarrow A[p..k-1]$ vero
$L[i]$ $R[j]$

fine: $k = n+1$ $A[p..r] \rightarrow$ vero

Titolo nota 02/03/2016

$$T(n) = \begin{cases} \Theta(1) & \text{se } n \leq K \\ a\,T\left(\frac{n}{b}\right) + c\, f(n) \end{cases}$$

numero dei sottoproblemi risolti ricorsivamente

dimensione dei sottoproblemi

Combina

equazione di ricorrenza

$$T(n) = \begin{cases} \Theta(1) & \text{se } n = 1 \\ \underbrace{2\,T\left(\frac{n}{2}\right)}_{\text{Risolvi}} + \underbrace{\Theta(n)}_{\text{Combina}} + \Theta(1) \end{cases}$$

divide

$n = 2^i$

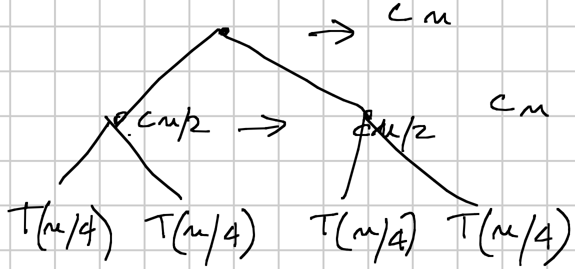
$T(1) = c_0$

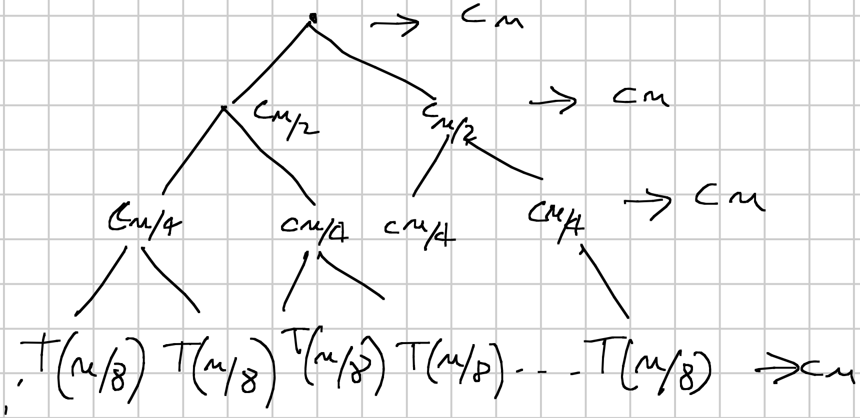
$T(n) = 2T\left(\frac{n}{2}\right) + cn$

$T(n/2) = 2T\left(\frac{n}{4}\right) + cn/2$

$T(n/4) = 2T\left(\frac{n}{8}\right) + cn/4$

cn





$\frac{n}{2^i} = 1 \qquad n = 2^i$

$\boxed{\Theta(n \log n)}$

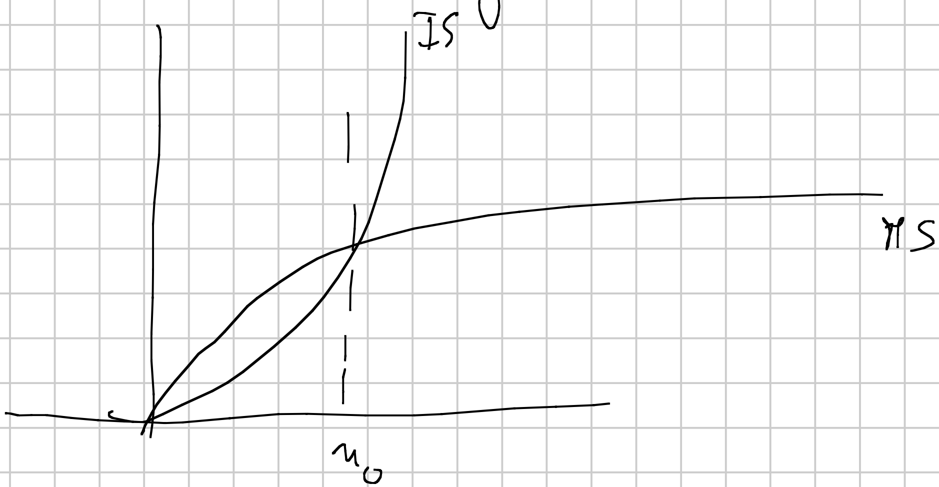
$i = \log_2 n$

$\boxed{T(n) = cn \cdot i = cn\log_2 n}$

$\boxed{T\left(\frac{n}{2^i} = 1\right)}$

SORTING

		tempo	spazio
Insertion Sort	IS	$O(n^2)$	$\Theta(1)$
Selection Sort	SS	$\Theta(n^2)$	$\Theta(1)$
Merge Sort	MS	$\Theta(n \log n)$	$\Theta(n)$



possiamo ancora migliorare?

$\Theta(n)$?

Titolo nota 02/03/2016

$n = 10^7$

Intel 7 (2013) 190·000 MIPS

COMMODORE 64 (1982) 0.5 MIPS

InsertionSort $= n^2$

MergeSort $= n\log n$

Tempo su INTEL usando IS : $\dfrac{n^2 \text{ operazioni}}{n \text{ operaz al sec}} = \dfrac{(10^7)^2}{190\cdot000 . 10^6} \simeq 526$ secondi $\simeq 9$ minuti

Tempo su COMMODORE usando MS : $\dfrac{n \log n \text{ op.}}{n. \text{op al sec}} = \dfrac{10.^7 \log 10}{0.5\ 10^6} = \dfrac{10^7 . 23}{0.5 . 10^6} = 460$ sec

LA POTENZA DI UN ALG. EFFICIENTE

$n = 10^8$

$T_{IS \text{ su INTEL}} = 15$ ore

$T_{MS \text{ su COMMODORE}} = 1$ ora e $1/2$

Titolo nota 02/03/2016

NOTAZIONI ASINTOTICHE

O , Ω , Θ , o , ω

O grande
big O
limiti sup

omega
limiti inferiori

Theta
limiti esatti

o piccolo
limiti superiori che non si raggiungono

ω piccolo
limiti inf. che non si raggiungono

Donald Knuth: The art of Computer Programming

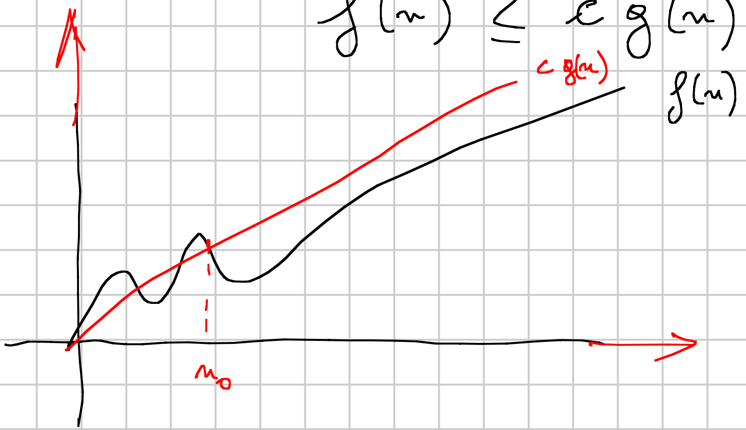
4 volumi

O

$$O(g(n)) = \{ f(n) : \exists\ c, n_0 \text{ t.c. } \forall n \geq n_0 \quad f(n) \leq c\, g(n) \}$$

$f(n) \in O(g(n))$

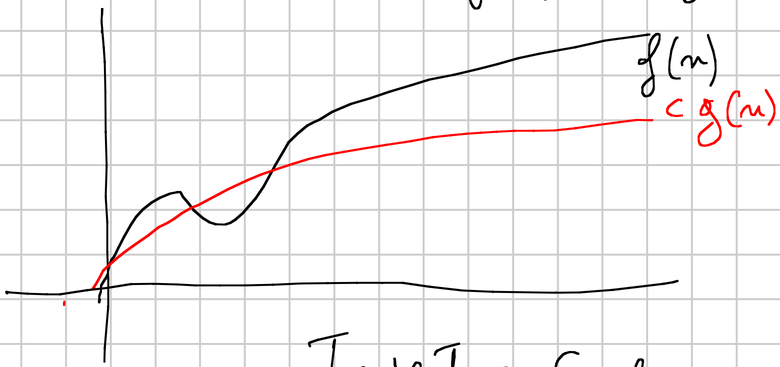
$f(n) = O(g(n))$



Titolo nota 02/03/2016

Ω limite inferiore

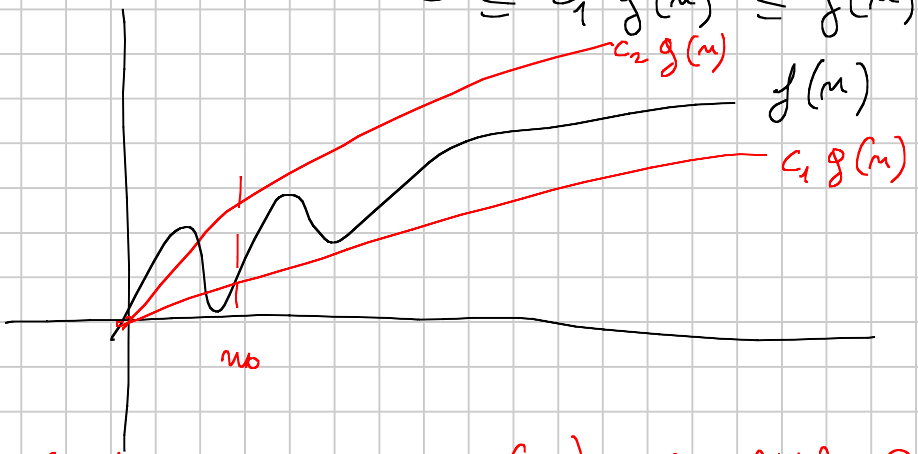
$$\Omega(g(n)) = \{ f(n) : \exists\, c, n_0 \text{ t.c } \forall n \geq n_0 \ \ f(n) \geq c\, g(n) \}$$



$$\text{InsertionSort} = \Omega(n)$$

Θ limite stretto

$$\Theta(g(n)) = \{ f(n) : \exists\ c_1, c_2, n_0 \text{ positive t.c } \forall n \geq n_0 \quad 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \}$$



f(n) cresce come g(n) a meno di fattori costanti

Titolo nota 02/03/2016

$$f(n) = 3n^2 \log n + 2n^4 = \Theta(n^4)$$

$$= 3\underline{n^3 \log^2 n} + 4n\log^3 n = \Theta(n^3 \log^2 n)$$

$$f(n) = 4n^{10} + 5n^2 + 6n - 2 = \Theta(n^{10})$$

se $f(n) = O(g(n)) \Rightarrow g(n) = O(f(n))$??

$$f(n) = O(g(n)) \Rightarrow g(n) = \Omega(f(n))$$

Titolo nota 02/03/2016

$$f(n) = a\,n^2 \log^3 n$$

$= \boxed{\Theta}(n^2 \log^3 n)$ sì

$= O(n^2)$ no

$= O(n^3)$ sì

$= O(n^2 \log^3 n)$ sì

$= O(n^2 \log^4 n)$ sì

$= O(n \log^3 n)$ no

$= \Omega(n \log^3 n)$ sì

$= \Omega(n^2)$ sì

se $f(n)$ è $\Theta(g(n))$

$$\Updownarrow$$

$$\begin{cases} f(n) \text{ è } O(g(n)) \\ f(n) \text{ è } \Omega(g(n)) \end{cases}$$

Titolo nota 02/03/2016

$$\underline{o(g(n))} = \{ f(n) \mid \forall c>0, \exists n_0 : \forall n \geq n_0 \quad 0 \leq f(n) < c\, g(n) \}$$

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

limite sup. che non si raggiunge

$$f(n) \text{ è } o(g(n))$$

$$n^2 = o(n^2 \log n)$$

$$\log^k n = o(n^{\varepsilon})$$

$$\underline{\omega(g(n))} = \text{— — — —}$$

$f(n) = \omega(g(n))$ allora $g(n) = o(f(n))$

Titolo nota

02/03/2016

Titolo nota

02/03/2016

Titolo nota

02/03/2016

Titolo nota 02/03/2016

Titolo nota

02/03/2016

Titolo nota

02/03/2016

Titolo nota

02/03/2016

02/03/2016

Titolo nota

Titolo nota

02/03/2016

02/03/2016

Titolo nota

Titolo nota

02/03/2016