Analisi di funzioni

(Si può fare meglio di così?)

Emanuele Nardi

Compilato il 29 giugno 2019
v1.0.0

Abbiamo concluso la prima parte delle lezioni che ci introduceva alle equazioni di ricorrenza, ora andremo a capire i fondamenti matematici che ci permetteranno di analizzare una famiglia più grande di equazioni di ricorrenza.

0.1 Proprietà della notazione asintotica

0.1.1 regola generale ↑

$$\boxed{f(n) = a_k n^k + a_{k-1} n^{k-1} + \dots a_1 n + a_0, a_k > 0 \Rightarrow f(n) = \Theta(n^k)}$$

Limite superiore: $\exists c > 0, \exists m \geqslant 0 : f(n) \leqslant cn^k, \forall n \geqslant m$

$$\begin{aligned}
f(n) &= a_k n^k + a_{k-1} n^{k-1} + \dots + a_1 n + a_0 && \text{↓ } a_k > 0 \text{ per def., rendo gli altri positivi} \\
&\leqslant a_k n^k + |a_{k-1}| n^{k-1} + \dots + |a_1| n + |a_0| && \text{↓ } \forall n \geqslant 1, \text{ elevo tutte le potenze a } k \\
&\leqslant a_k n^k + |a_{k-1}| n^k + \dots + |a_1| n^k + |a_0| n^k && \text{↓ raccolgo per } n^k \\
&= (a_k + |a_{k-1}| + \dots + |a_1| + |a_0|)\, n^k && \text{↓ esiste...} \\
&\overset{?}{\leqslant} cn^k
\end{aligned}$$

che è vera per $c \geqslant (|a_k| + |a_{k-1}| + \dots + |a_1| + |a_0|)$ (il coefficiente) e per $m = 1$.

Limite inferiore: $\exists d > 0, \exists m \geqslant 0 : f(n) \geqslant dn^k, \forall n \geqslant m$

$$\begin{aligned}
f(n) &= a_k n^k + a_{k-1} n^{k-1} + \dots + a_1 n + a_0 && \text{↓ normalizzo aggiungendo un segno negativo} \\
&\geqslant a_k n^k - |a_{k-1}| n^{k-1} - \dots - |a_1| n - |a_0| && \text{↓ } \forall n \geqslant 1, \text{ elevo alla } k-1 \\
&\geqslant a_k n^k - |a_{k-1}| n^{k-1} - \dots - |a_1| n^{k-1} - |a_0| n^{k-1} \\
&\overset{?}{\geqslant} dn^k
\end{aligned}$$

L'ultima equazione è vera se:

$$d \leqslant a_k - \frac{|a_{k-1}|}{n} - \frac{|a_{k-2}|}{n} - \dots - \frac{|a_1|}{n} - \frac{|a_0|}{n} > 0 \iff n > \frac{|a_{k-1}| + \dots + |a_0|}{a_k} = m$$

Costituito da una componente positiva (a_k) e da una componente negativa che per valori crescenti di n tende a 0.
Abbiamo dimostrato sia il limite superiore che quello inferiore, possiamo affermare che è un Θ.

Ad esempio $17n^3 - 47n^2 + 123n + 17 \Rightarrow \Theta(n^3)$. Oppure $2n^3 + 7 = \Theta(n^2)$.

0.1.2 Funzioni di costo particolari

La complessità di $f(n) = 5$ è pari a $\Theta(1)$, questa classe di complessità appartiene a quegli algoritmi che sono così ben congegnati che indipendentemente dalla dimensione dell'input ci mettono un tempo costante a risolvere il problema. Ad esempio richiedere il minimo in un vettore ordinato.

Dimostriamolo.

$$f(n) = 5 \geqslant c_1 n^0 \Rightarrow c_1 \leqslant 5$$
$$f(n) = 5 \leqslant c_2 n^0 \Rightarrow c_2 \geqslant 5$$
$$f(n) = \Theta(n^0) = \Theta(1)$$

La complessità di $f(n) = 5 + \sin(n)$ è pari a $\Theta(1)$, in quanto oscilla fra 1 e -1.

0.1.3 Proprietà delle notazioni

Domanda orale

Definizione 0.1 (Dualità)**.**

$$f(n) = \mathcal{O}(g(n)) \Leftrightarrow g(n) = \Omega(f(n))$$

La dimostrazione avviene tramite passaggi algebrici:

$$\begin{aligned} f(n) = \mathcal{O}(g(n)) &\Leftrightarrow f(n) \leqslant \boldsymbol{c}\ g(n), \forall n \geqslant m && \text{ribalto la disequazione, } c > 0 \\ &\Leftrightarrow g(n) \leqslant \tfrac{\mathbf{1}}{\boldsymbol{c}}\ f(n), \forall n \geqslant m && \text{rinomino } \tfrac{1}{c} \text{ a } c' \\ &\Leftrightarrow g(n) = \boldsymbol{c'}\ f(n), \forall n \geqslant m, c' = \tfrac{1}{c} && \text{passo alla notazione dei quantificatori} \\ &\Leftrightarrow g(n) = \Omega(f(n)) \end{aligned}$$

Definizione 0.2 (Eiminazione delle costanti)**.**

$$f(n) = \mathcal{O}(g(n)) \Leftrightarrow af(n) = \mathcal{O}(g(n)), \forall a > 0$$
$$f(n) = \Omega(g(n)) \Leftrightarrow af(n) = \Omega(g(n)), \forall a > 0$$

La dimostrazione avviene sempre tramite semplici passaggi algebrici:

$$\begin{aligned} f(n) = \mathcal{O}(g(n)) &\Leftrightarrow f(n) \leqslant cg(n), \forall n \geqslant m && \text{Introduco una costante } a \\ &\Leftrightarrow f(n) \leqslant acg(n), \forall n \geqslant m, \forall a \geqslant 0 && \text{raccolgo } ac \text{ sotto un'unica costante } c' \\ &\Leftrightarrow f(n) \leqslant c'g(n), \forall n \geqslant m, c' = ac > 0 && \text{per definizione} \\ &\Leftrightarrow af(n) = \mathcal{O}(g(n)) \end{aligned}$$

Ad esempio $2 \log n = \Theta(\log n)$, ignoriamo quindi le costanti numeriche.

Definizione 0.3 (Sommatoria (sequenza di algoritmi))**.**

$$\begin{cases} f_1(n) = \mathcal{O}(g_1(n)) \\ f_2(n) = \mathcal{O}(g_2(n)) \end{cases} \Rightarrow f_1(n) + f_2(n) = \mathcal{O}(\mathsf{max}(g_1(n), g_2(n)))$$
$$\begin{cases} f_1(n) = \Omega(g_1(n)) \\ f_2(n) = \Omega(g_2(n)) \end{cases} \Rightarrow f_1(n) + f_2(n) = \Omega(\mathsf{min}(g_1(n), g_2(n)))$$

La dimostrazione.

$$\begin{aligned} f_1(n) = \mathcal{O}(g_1(n)) \wedge f_2 n = \mathcal{O}(g_2(n)) &\Rightarrow \quad \text{per definizione} \\ f_1(n) \leqslant c_1 g_1(n) \wedge f_2(n) \leqslant c_2 g_2(n) &\Rightarrow \quad \text{raccolgo} \\ f_1(n) + f_2(n) \leqslant c_1 g_1(n) + c_2 g_2(n) &\Rightarrow \quad \text{pre definizione} \\ f_1(n) + f_2(n) \leqslant \max\{c_1, c_2\}(2 \cdot \max(g_1(n), g_2(n)) &\Rightarrow \\ f_1(n) + f_2(n) = \mathcal{O}(g_1(n) + g_2(n)) & \end{aligned}$$

Ad esempio se ripeto due volte un algoritmo lineare, l'algoritmo risultante sarà comunque lineare. Mentre se ho un algoritmo lineare ed un algoritmo quadratico e la complessità risultante è una combinazione delle due, la complessità totale sarà quindi quadratica.

Definizione 0.4 (Cicli annidati)**.** *Se $f_1(n)$ viene ripetuto $f_2(n)$ volte, allora $f_1(n) \cdot f_2(n)$ è limitato superiormente dalla complessità maggiore fra i due e limitato inferiormente dalla complessità minore fra i due.*

$$\begin{cases} f_1(n) = \mathcal{O}(g_1(n)) \\ f_2(n) = \mathcal{O}(g_2(n)) \end{cases} \Rightarrow f_1(n) \cdot f_2(n) = \mathcal{O}(\max(g_1(n) \cdot g_2(n)))$$

$$\begin{cases} f_1(n) = \Omega(g_1(n)) \\ f_2(n) = \Omega(g_2(n)) \end{cases} \Rightarrow f_1(n) \cdot f_2(n) = \Omega(\min(g_1(n) \cdot g_2(n)))$$

La dimostrazione è molto semplice.

$$\begin{aligned} f_1(n) = \mathcal{O}(g_1(n)) \wedge f_2 n = \mathcal{O}(g_2(n)) &\Rightarrow \quad \text{per definizione} \\ f_1(n) \leqslant c_1 g_1(n) \wedge f_2(n) \leqslant c_2 g_2(n) &\Rightarrow \quad \text{raccolgo, } c_1 c_2 > 0 \\ f_1(n) \cdot f_2(n) \leqslant c_1 c_2 g_1(n) g_2(n) & \end{aligned}$$

Ad esempio se ripeto n volte un algoritmo di costo $\log n$, quell'algoritmo avrà un costo totale di $n \log n$.

Definizione 0.5 (Simmetria)**.** *Se $f(n)$ è limitata superiormente ed inferiormente da $g(n)$, allora anche $g(n)$ è limitata inferiormente e superiormente da $f(n)$.*

$$f(n) = \Theta(g(n)) \Leftrightarrow g(n) = \Theta(f(n))$$

Vuol dire semplicemente che $2n^2 + 7 = \Theta(n^2) \Leftrightarrow n^2 = \Theta(2n^2 + 7)$.
La dimostrazione avviene tramite la proprietà di dualità:

$$\begin{aligned} f(n) = \Theta(g(n)) \quad &\Rightarrow \quad f(n) = \mathcal{O}(g(n)) \quad \Rightarrow \quad g(n) = \Omega(f(n)) \\ f(n) = \Theta(g(n)) \quad &\Rightarrow \quad f(n) = \Omega(g(n)) \quad \Rightarrow \quad g(n) = \mathcal{O}(f(n)) \end{aligned}$$

Definizione 0.6 (Transitività)**.** *Se $f(n)$ è limitata superiormente da $g(n)$, e $g(n)$ è limitata superiormente da $h(n)$, allora $f(n)$ è limitata superiormente da $h(n)$.*

$$\begin{cases} f(n) = \mathcal{O}(g(n)) \\ g(n) = \mathcal{O}(h(n)) \end{cases} \Rightarrow f(n) = \mathcal{O}(h(n))$$

La dimostrazione è banale:

$$\begin{aligned} f(n) = \mathcal{O}(g(n)) \wedge g(n) = \mathcal{O}(h(n)) &\Rightarrow \quad \text{elimino i quantificatori} \\ f(n) \leqslant c_1 g(n) \wedge g(n) \leqslant c_2 h(n) &\Rightarrow \quad \text{sostituisco } g(n) \text{ con } c_2 h(n) \\ f(n) \leqslant c_1 c_2 h(n) &\Rightarrow \quad c_1 c_2 > 0, \text{ elimino la costante} \\ f(n) = \mathcal{O}(h(n)) & \end{aligned}$$

Altre funzioni di costo

Vogliamo provare che $\boxed{\log n = \mathcal{O}(n)}$

Dimostriamo per induzione che $\exists c > 0, \exists m \geqslant 0 : \log n \leqslant cn, \forall n \geqslant m$.

- **caso base** ($n = 1$):

$$\begin{aligned} \log n &\leqslant cn && \text{(} n = 1 \text{)} \\ \log 1 &\leqslant c \cdot 1 && \text{(semplifico)} \\ 0 &\leqslant c \end{aligned}$$

- **ipotesi induttiva**: $\log k \leqslant ck, \forall k \leqslant n$
- **passo induttivo** dimostriamo la proprietà per $n + 1$:

$$\begin{aligned} \log(n+1) \leqslant \log(n+n) &= \log 2n \quad \forall n \geqslant 1 && \text{(} \log ab = \log a + \log b \text{)} \\ &= \log 2 + \log n && \text{(} \log_2 2 = 1 \text{)} \\ &= 1 + \log n && \text{(per ipotesi induttiva)} \\ &\leqslant 1 + cn && \text{(obiettivo)} \\ &\overset{?}{\leqslant} c(n+1) && \text{(metto a confronto)} \\ 1 + cn &\leqslant c(n+1) && \text{(moltiplico)} \\ 1 + cn &\leqslant cn + c) && \text{(semplifico)} \\ 1 &\leqslant c \end{aligned}$$

Classificazione delle funzioni

É possibile trarre un ordinamento dalle principali espressioni, estendendo le relazioni che abbiamo dimostrato fino ad ora. Per ogni $r < s, h < k, a < b$:

$$\mathcal{O}(1) \subset \mathcal{O}(\log^r n) \subset \mathcal{O}(\log^s n) \subset \mathcal{O}(n^h) \subset \mathcal{O}(n^h \log^r n) \subset \mathcal{O}(n^h \log^s n) \subset \mathcal{O}(n^k) \subset \mathcal{O}(a^n) \subset \mathcal{O}(b^n)$$

Ricordati che $\log^r(n) = (\log n)^r$. Non è detto che gli esponenti siano interi. $\sqrt[1000]{n}$ cresce più velocemente di $\log^2 n$. n cresce meno velocemente di $n \log n$, il quale a sua volta cresce meno velocemente di $n \log^2 n$.

0.2 Analisi per livelli

Nell'analisi per livelli (o dell'albero di ricorsione) andiamo sostanzialmente a "srotolare" la ricorrenza in un albero, i cui nodi rappresentano i casti ai vari livelli della ricorsione.

0.2.1 Esempi di analisi per livelli

Analisi per livelli dell'algoritmo della ricerca binaria

Equazione di ricorrenza della ricerca binaria:

$$T(n) = \begin{cases} T(n/2) + b & n > 1 \\ 1 & n \leq 1 \end{cases}$$

É possibile risolvere questa ricorrenza nel seguente modo:

$$
\begin{aligned}
T(n) &= b + T(\tfrac{n}{2}) && \text{↓ } T\left(\tfrac{n}{2}\cdot\tfrac{1}{2}\right)+b\\
&= b + b + T(\tfrac{n}{4}) && \text{↓ } T\left(\tfrac{n}{4}\cdot\tfrac{1}{2}\right)+b\\
&= b + b + b + T(\tfrac{n}{8}) && \text{↓ } \textit{svolgo } \log n \textit{ operazioni}\\
&= \cdots\\
&= \underbrace{b + b + \ldots + b}_{\log n} + T(1) && \text{↓ } \textit{semplifico}\\
&= b\log n + T(1)\\
&= \Theta(\log n) && \text{↓ } \textit{eliminazione delle costanti}
\end{aligned}
$$

Abbiamo assunto per semplicità che $n = 2^k$, ovvero che $k = \log n$.

Analisi per livelli del primo tentativo moltiplicazione Karatsuba

Equazione di ricorrenza:

$$
T(n) = \begin{cases} 4T(n/2) + n & n > 1 \\ 1 & n \leqslant 1 \end{cases}
$$

É possibile risolvere questa ricorrenza nel modo seguente:

$$
\begin{aligned}
T(n) &= n + 4T(n/2) && \text{↓ } 4T\left(\tfrac{n}{2}\cdot\tfrac{1}{2}\right)+\tfrac{n}{2}\\
&= n + 4\left(4T\left(\frac{n}{2}\cdot\frac{1}{2}\right)+\frac{n}{2}\right) && \text{↓ } \textit{semplifico}\\
&= n + 4n/2 + 16T(n/4) && \text{↓ } 4T\left(\tfrac{n}{4}\cdot\tfrac{1}{2}\right)+\tfrac{n}{4},\ 4n/2 = 2n\\
&\quad n + 2n + 16\left(4T\left(\frac{n}{4}\cdot\frac{1}{2}\right)+\frac{n}{4}\right) && \text{↓ } \textit{semplifico}\\
&= n + 2n + 16n/4 + 64T(n/8) && \text{↓ } \textit{svolgo } \log n \textit{ operazioni}\\
&= \cdots\\
&= n + 2n + 4n + 8n + \cdots + 2^{\log n - 1}n + 4^{\log n}T(1) && \text{↓ } \textit{raccolgo}\\
&= n\sum_{j=0}^{\log n - 1} 2^j + 4^{\log n}
\end{aligned}
$$

Ciò che abbiamo ottenuto è una forma chiusa, non più un'equazione di ricorrenza, ma non è ancora nella sua forma definitiva, dobbiamo ancora trattarla.

$$
\begin{aligned}
T(n) &= n\sum_{j=0}^{\log n - 1} 2^j + 4^{\log n} && \text{↓ } \textit{Applico } \forall x \neq 1 : \textstyle\sum_{j=0}^{k} x^j = \frac{x^{k+1}-1}{x-1},\ \textit{dove } k = \log n - 1\\
&= n \cdot \frac{2^{\log n} - 1}{2 - 1} + 4^{\log n} && \text{↓ } 2^{\log n} = n^{\log_2 2} = n^1\\
&= n(n-1) + 4^{\log n} && \text{↓ } \textit{moltiplico: } n(n-1) = n^2 - n\\
&\quad && \textit{semplifico: } 4^{\log n} = n^{\log 4} = n^2\\
&= n^2 - n + n^2 && \text{↓ } \textit{raccolgo } n^2\\
&= 2n^2 - n && \text{↓ } \textit{regola generale } \uparrow\\
&= \Theta(n^2)
\end{aligned}
$$

Possiamo concludere che $T(n) = n + 4T(n/2) = \Theta(n^2)$.

Modifichiamo l'esempio precendente

Esaminiamo la seguente equazione di ricorrenza:

$$T(n) = \begin{cases} 4T(n/2) + n^3 & n > 1 \\ 1 & n \leqslant 1 \end{cases}$$

Per semplicità consideriamo $n = 2^\ell$.

livello	dim.	costo chiam.	no. chiamate	costo livello
0	n	n^3	1	n^3
1	$n/2$	$(n/2)^3$	4	$4(n/2)^3$
2	$n/4$	$(n/4)^3$	16	$16(n/4)^3$
3	$n/8$	$(n/8)^3$	64	$64(n/8)^3$
...	...	...	...	...
i	$n/2^i$	$(n/2^i)^3$	4^i	$4^i(n/2^i)^3$
...	...	...	...	...
$\ell - 1$	$n/2^{\ell-1}$	$(n/2^{\ell-1})^3$	$4^{\ell-1}$	$4^{\ell-1}(n/2^{\ell-1})^3$
$\ell = \log n$	1	$T(1)$	$4^{\log n}$	$4^{\log n}$

Sommando il costo di tutti i livelli otteniamo:

$$\begin{aligned}
T(n) &= \sum_{i=0}^{\log n - 1} 4^i \cdot \frac{n^3}{2^{3i}} + 4^{\log n} && \text{semplifico: } 4^i = 2^{2i} \\
&= n^3 \sum_{i=0}^{\log n - 1} \frac{2^{2i}}{2^{3i}} + 4^{\log n} && \text{semplifico: } \tfrac{2^{2i}}{2^{3i}} = \left(\tfrac{1}{2}\right)^i \\
&= n^3 \sum_{i=0}^{\log n - 1} \left(\frac{1}{2}\right)^i + 4^{\log n} && \text{cambio di base, } 4^{\log n} = n^{\log 4} = n^2 \\
&= n^3 \sum_{i=0}^{\log n - 1} \left(\frac{1}{2}\right)^i + n^2 && \text{estensione della sommatoria ad } \infty \\
&\leqslant n^3 \sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^i + n^2 && \text{Serie geometrica infinita decrescente:} \\
&= n^3 + \frac{1}{1 - \frac{1}{2}} + n^2 && \forall x, |x| < 1 : \textstyle\sum_{i=0}^{\infty} x^i = \frac{1}{1-x}, \text{ dove } x = \frac{1}{2} \\
&= 2n^3 + n^2 && \text{semplifico: } \tfrac{1}{1-\frac{1}{2}} = 2
\end{aligned}$$

Abbiamo dimostrato che $T(n) \leqslant 2n^3 + n^2$, possiamo quindi affermare che $T(n) = \mathcal{O}(n^3)$, ma non possiamo affermare che $T(n) = \Theta(n^3)$ poiché abbiamo dimostrato solo un limite superiore.

Suggerimento. *Tutte le volte che notiamo in un'equazione di riccorrenza una parte polinomiale, come ad esempio n^3, possiamo dire con certezza che $\Omega(n^3)$.*

Ora possiamo affermare che $T(n) = \Theta(n^3)$.

Modifichiamo l'esempio precendente ulteriormente

Esaminiamo la seguente equazione di ricorrenza:

$$T(n) = \begin{cases} 4T(n/2) + n^3 & n^2 > 1 \\ 1 & n \leqslant 1 \end{cases}$$

Cambia il costo della chiamata, che passa dall'essere n^3 all'essere n^2. Di conseguenza cambia anche il costo del livello.

Livello	Dim.	Costo chiam.	no. chiamate	Costo livello
0	n	n^3	1	n^2
1	$n/2$	$(n/2)^2$	4	$4(n/2)^2$
2	$n/4$	$(n/4)^2$	16	$16(n/4)^2$
3	$n/8$	$(n/8)^2$	64	$64(n/8)^2$
...	...	...	...	...
i	$n/2^i$	$(n/2^i)^2$	4^i	$4^i(n/2^i)^2$
...	...	...	...	...
$\ell-1$	$n/2^{\ell-1}$	$(n/2^{\ell-1})^2$	$4^{\ell-1}$	$4^{\ell-1}(n/2^{\ell-1})^2$
$\ell=\log n$	1	$T(1)$	$4^{\log n}$	$4^{\log n}$

Sommando il costo di tutti i livelli otteniamo:

$$
\begin{aligned}
T(n) &= \sum_{i=0}^{\log n-1} \frac{n^2}{2^{2i}} \cdot 4i + 4^{\log_2 n} \\
&= n^2 \sum_{i=0}^{\log n-1} \frac{2^{2i}}{2^{2i}} + n^2 \\
&= n^2 \sum_{i=0}^{\log n-1} 1 + n^2 \\
&= n^2 \log n + n^2 \\
&= \Theta(n^2 log n)
\end{aligned}
$$

semplifico: $4^i = 2^{2i}$, $4^{\log n} = n^{\log_2 4} = n^2$

semplifico: $\frac{2^{2i}}{2^{2i}} = 1$

svolgo la sommatoria: $\sum_{i=0}^{\log n-1} 1 = \log n$

regola generale ↑

Conclusioni

Per riassumere n ha prodotto n^2, n^2 ha prodotto $n^2 \log n$ ed n^3 ha prodotto n^3. Quando avremo a disposizione lo strumento "master theorem" riusciremo semplicemente guardando l'equazione di ricorrenza a capire quale complessità l'equazione di ricorrenza produce.

0.3 Metodo di sostituzione

Vediamo un altro meccanismo poiché il metodo precendente può non esserci di aiuto. Il metodo di sostituzione è un metodo in cui si cerca di "indovinare" (*guess*) una soluzione in base alla propria esperienza, e si dimostra che questa soluzione è corretta tramite induzione.

Primo esercizio

$$
T(n) = \begin{cases} T(\lfloor n/2 \rfloor) + n & n > 1 \\ 1 & n \leqslant 1 \end{cases}
$$

Risolvendolo tramite il metodo precendente otteniamo:

$$
\begin{aligned}
T(n) &= n \sum_{i=0}^{\log n} \left(\frac{1}{2}\right)^i && \text{estensione della sommatoria ad } \infty \\
&\leqslant n \sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^i && \text{Serie geometrica decrescente infinita: } \forall x, |x| < 1 : \textstyle\sum_{i=0}^{\infty} x^i = \frac{1}{1-x}, \text{ dove } x = \frac{1}{2} \\
&\leqslant n \frac{1}{1-\frac{1}{2}} && \frac{1}{1-\frac{1}{2}} = 2 \\
&= 2n
\end{aligned}
$$

Sapendo già il risultato proviamo – per tentativi – a dimostrare che $\boxed{T(n) = \mathcal{O}(n)}$

Limite superiore: Dobbiamo dimostrare che $\exists c > 0, \exists m \geqslant 0 : T(n) \leqslant cn, \forall n \geqslant m$

- **caso base** dimostriamo $T(1)$:

$$T(1) = 1 \overset{?}{\leqslant} c \cdot 1 \iff c \geqslant 1$$

- **ipotesi induttiva** $\forall k < n : T(n) \leqslant ck$, ossia affermiamo che per tutti i valori più piccoli di n la dimostrazione è già stata fatta.
- **ipotesi induttiva** dimostriamo la disequazione per $T(n)$:

$$
\begin{aligned}
T(n) &= T(\lfloor \tfrac{n}{2} \rfloor) + n && \text{sost. ip. ind. con } k = \lfloor \tfrac{n}{2} \rfloor \\
&\leqslant c\lfloor \tfrac{n}{2} \rfloor + n && \text{semplifico l'intero inferiore} \\
&\leqslant c\tfrac{n}{2} + n && \text{raccolgo } n \\
&= (\tfrac{c}{2} + 1)n && \text{obiettivo} \\
&= (\tfrac{c}{2} + 1)n \overset{?}{\leqslant} cn && \text{semplifico} \\
&\Leftrightarrow \tfrac{c}{2} + 1 \leqslant c \\
&\Leftrightarrow c \geqslant 2
\end{aligned}
$$

Abbiamo quindi provato che $T(n) \leqslant cn$, con due diversi valori della nostra costante c: nel caso base è risultata $c \geqslant 1$, mentre nel passo induttivo è risultata pari a $c \geqslant 1$. Possiamo accettare solo un valore di c che *rispetti entrambe le disequazioni*, in questo caso è $c = 2$.

Quanto abbiamo dimostrato vale per $n = 1$ e per tutti i valori di n successivi, di conseguenza $m = 1$.

Al solo scopo didattico proviamo passo passo anche il limite inferiore, ossia che $\boxed{T(n) = \Omega(n)}$

Limite inferiore: Dobbiamo dimostrare che $\exists d > 0, \exists m \geqslant 0 : T(n) \geqslant dn, \forall n \geqslant m$.

Suggerimento. *Usiamo la costante d al solo scopo di non confonderla la costante c usata nella dimostrazione precendente.*

- **caso base** $T(1)$:

$$T(1) = 1 \overset{?}{\geqslant} d \cdot 1 \iff d \leqslant 1$$

- **ipotesi induttiva** $\forall k < n : T(k) \geqslant ck$, ossia affermiamo che per tutti i valori più piccoli di n la mia dimostrazione è già stata fatta.

- **ipotesi induttiva** dimostriamo la disequazione per $T(n)$

$$
\begin{aligned}
T(n) &= T(\lfloor \tfrac{n}{2} \rfloor) + n && \text{sost. ip. ind. con } d = \tfrac{n}{2} \\
&\geqslant d\lfloor \tfrac{n}{2} \rfloor + n && \text{semplifico l'intero inferiore} \\
&\geqslant d\tfrac{n}{2} - 1 + n && \text{raccolgo } n \\
&= \left(\tfrac{d}{2} - \tfrac{1}{n} + 1\right) n && \text{obiettivo} \\
&= \left(\tfrac{d}{2} - \tfrac{1}{n} + 1\right) n \overset{?}{\geqslant} dn && \text{semplifico} \\
&\Leftrightarrow \tfrac{d}{2} - \tfrac{1}{n} + 1 \geqslant n \\
&\Leftrightarrow d \leqslant 2 - \tfrac{2}{n}
\end{aligned}
$$

  Abbiamo quindi provato che $T(n \geqslant dn$, con due diversi valori della nostra costante d: nel caso base è risultata $d \leqslant 1$, mentre nel passo induttivo è risultata pari a $d \leqslant 2 - \frac{2}{n}$. Possiamo accettare solo un valore di d che *rispetti entrambe le disequazioni* per ogni valore di $n \geqslant 1$, in questo caso è $d = 1$.

  Quanto abbiamo dimostrato vale per $n = 1$ e per tutti i valori di n successivi, di conseguenza $m = 1$.

  Per concludere abbiamo provato che $T(n) = T(\lfloor \frac{n}{2} \rfloor) + n$ è limitata sia superiormente $T(n) = \mathcal{O}(n)$, sia inferiormente $T(n) = \Omega(n)$, possiamo affermare quindi con assoluta certezza che $T(n) = \Theta(n)$.

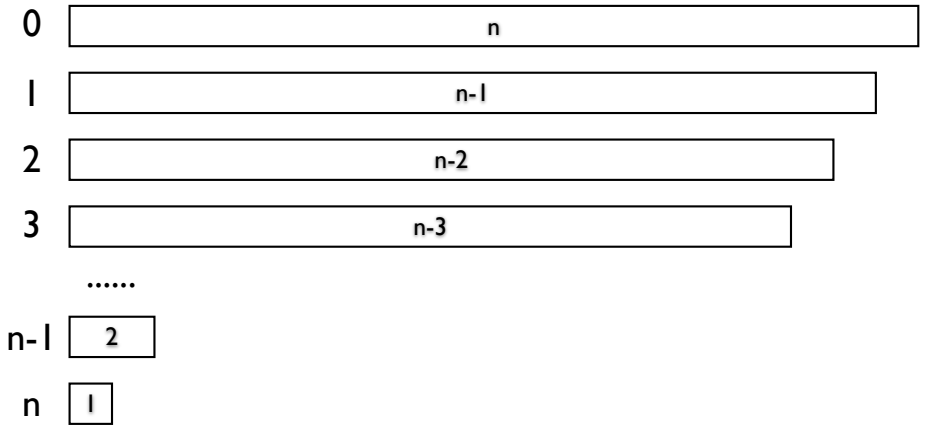
Nota che avremmo potuto evitare la dimostrazione del limite inferiore osservando che

$$T(n) = T(\lfloor \tfrac{n}{2} \rfloor) + n \geqslant n \overset{?}{\geqslant} dn$$

l'ultima disequazione risulta vera per $d \leqslant 1$, la quale è una condizione identica a quella del caso base. Nota che non abbiamo fatto nemmeno ricordo all'ipotesi induttiva.

Cosa succede se si sbaglia l'intuizione

$$T(n) = \begin{cases} T(n-1) + n & n > 1 \\ 1 & n \leqslant 1 \end{cases}$$



Possiamo rappresentare la funzione nel seguente modo:

$$T(n) = \sum_{i=1}^{n} i = \frac{n(n+1)}{2} = \Theta(n^2)$$

Effettuiamo un tentativo e dimostriamo che $\boxed{T(n) = \mathcal{O}(n)}$

Limite superiore: Dobbiamo dimostrare che $\exists c > 0, \exists m \geqslant 0 : T(n) \leqslant cn, \forall n \geqslant m$.

- **caso base** lo saltiamo perchè vedremmo subito che è sbagliato.
- **ipotesi induttiva** $\forall k < n : T(k) \leqslant ck$.
- **passo induttivo** dimostriamo la disequazione per $T(n)$:

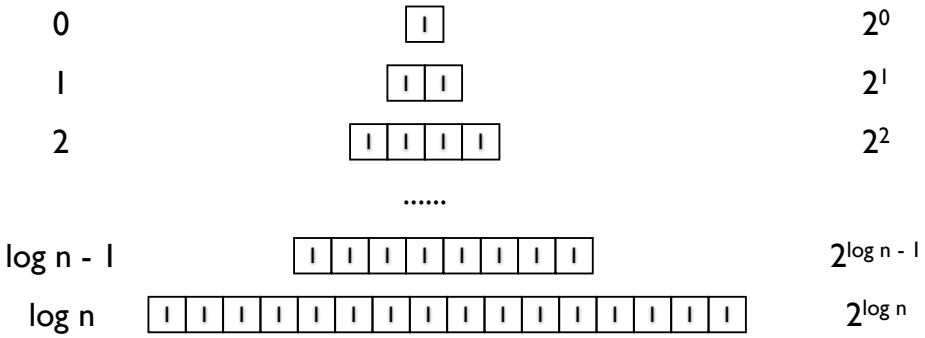
$$
\begin{aligned}
T(n) &= T(n-1) + n && \text{sost. ip. ind. con } k = n-1 \\
&= c(n-1) + n && \text{moltiplico} \\
&= cn - c + n && \text{raccolgo } n \\
&= (c+1)n - c && \text{rimuovo l'elemento negativo} \\
&= (c+1)n && \text{obiettivo} \\
&= (c+1)n \overset{?}{\leqslant} cn && \text{semplifico} \\
&\Leftrightarrow c + 1 \leqslant c
\end{aligned}
$$

Possiamo notare che l'ultima disequazione risulta impossibile.

Difficoltà matematica

$$
T(n) = \begin{cases} T(\lfloor \frac{n}{2} \rfloor) + T(\lceil \frac{n}{2} \rceil) + 1 & n > 1 \\ 1 & n \leqslant 1 \end{cases}
$$

Nota. *É possibile ottenere questa equazione di riccorrenza dall'algoritmo che calcola il minimo di un vettore non ordinato in maniera ricorsiva.*



$$
T(n) = \sum_{i=0}^{\log n} 2^i = n + \frac{n}{2} + \frac{n}{4} + \cdots + 1 = \mathcal{O}(n)
$$

Effettuiamo un tentativo per $\boxed{T(n) = \mathcal{O}(n)}$

- **ipotesi induttiva**: $\forall k < n : T(k) \leqslant ck$.

- **passo induttivo**: dimostriamo la disequazione per $T(n)$:

$$\begin{aligned} T(n) &= T(\lfloor \tfrac{n}{2} \rfloor) + T(\lceil \tfrac{n}{2} \rceil) + 1 && \text{sost. ip. ind.} \\ &= c(\lfloor \tfrac{n}{2} \rfloor) + c(\lceil \tfrac{n}{2} \rceil) + 1 && \text{semplifichiamo, } \lfloor \tfrac{n}{2} \rfloor + \lceil \tfrac{n}{2} \rceil = n \\ &= cn + 1 && \text{obiettivo} \\ &= cn + 1 \overset{?}{\leqslant} cn && \text{semplifico} \\ &\Leftrightarrow 1 \leqslant 0 \end{aligned}$$

  Anche in questo caso notiamo che l'ultima disequazione risulta impossibile, ma – a differenza del caso precendente – non riusciamo a dimostrare il passo induttivo per un termine di ordine inferiore. Il tentativo risulta quindi errato.

  Proviamo quindi ad utilizzare un'ipotesi induttiva *più stretta*.

- **ipotesi induttiva più stretta**: $\exists c > 0, \exists m \geqslant 0 : T(n) \leqslant cn - b, \forall n \geqslant m$.

  Abbiamo introdotto una costante $b > 0$ nella nostra tesi, questa modifica ci permetterà di dimostrare correttamente il passo induttivo.

- **ipotesi induttiva** $\exists b > 0, \forall k < n : T(k) \leqslant ck - b$.

- **passo induttivo** dimostriamo la disequazione per $T(n)$:

$$\begin{aligned} T(n) &= T(\lfloor \tfrac{n}{2} \rfloor) + T(\lceil \tfrac{n}{2} \rceil) + 1 && \text{sost. ip. ind.} \\ &= c(\lfloor \tfrac{n}{2} \rfloor) - b + c(\lceil \tfrac{n}{2} \rceil) - b + 1 && \text{semplifichiamo} \\ &= cn - 2b + 1 && \text{obiettivo} \\ &= cn - 2b + 1 \overset{?}{\leqslant} cn - b && \text{semplifico} \\ &\Leftrightarrow -2b + 1 \leqslant -b \\ &\Leftrightarrow b \geqslant 1 \end{aligned}$$

- **caso base**

$$T(n) = 1 \overset{?}{\leqslant} c \cdot 1 - b \Leftrightarrow c \geqslant b + 1$$

  Per concludere abbiamo provato che $T(n) =\leqslant cn - b \leqslant cn$ con diversi valori delle costanti c e b, nel passo induttivo $\forall b \geqslant 1, \forall c$, nel caso base $\forall c \geqslant b + 1$. Una coppia di valori di b e c che rispettano queste disequazioni sono $b = 1$, $c = 2$.

  Questo vale per $n = 1$, e per tutti i valori di n successivi, quindi per $m = 1$.

  Abbiamo quindi provato che $T(n) = \mathcal{O}(n)$.

Dimostriamo il limite inferiore facendo un tentativo per $\boxed{T(n) = \Omega(n)}$

Dobbiamo dimostrare che $\exists d > 0, \exists m \geqslant 0 : T(n) \geqslant dn, \forall n \geqslant m$.

- **passo induttivo**

$$\begin{aligned} T(n) &= T(\lfloor \tfrac{n}{2} \rfloor) + T(\lceil \tfrac{n}{2} \rceil) + 1 && \text{sost. ip. ind.} \\ &\geqslant d(\lfloor \tfrac{n}{2} \rfloor) - b + d(\lceil \tfrac{n}{2} \rceil) - b + 1 && \text{semplifichiamo} \\ &= dn + 1 && \text{obiettivo} \\ &= dn + 1 \overset{?}{\leqslant} dn \end{aligned}$$

  L'ultima disequazione risulta vera $\forall d$.

- **caso base**

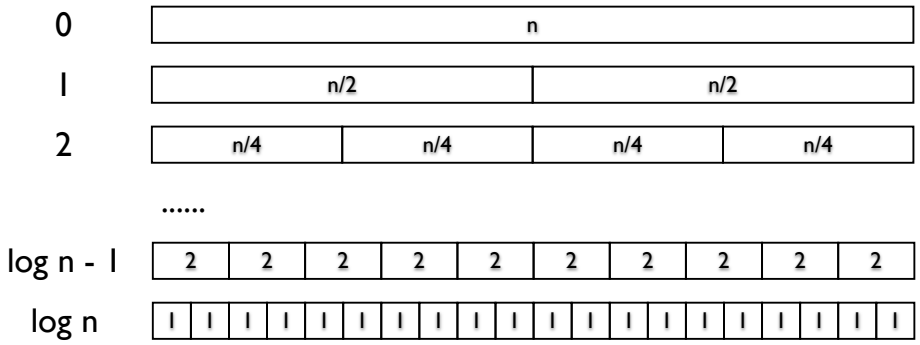
$$T(n) = 1 \geqslant d \cdot 1 \iff d \leqslant 1$$

Abbiamo quindi provato che $T(n) = \Omega(n)$

Problemi con i casi base

$$T(n) = \begin{cases} 2T(\lfloor \frac{n}{2} \rfloor) + n & n > 1 \\ 1 & n \leqslant 1 \end{cases}$$

Proviamo a visualizzarlo graficamente così:



Nota. *É molto simile all'equazione dell'algoritmo* mergeSort *che sappiamo avere una complessità di* $\mathcal{O}(n \log n)$.

Effettuiamo un tentativo per $\boxed{T(n) = \mathcal{O}(n \log n)}$

Dobbiamo dimostrare che $\exists c > 0, \exists m \geqslant 0 : T(n) \geqslant n \log n, \forall n \geqslant m$.

- **Ipotesi induttiva**: $\exists c > 0, \forall k < n : T(k) \leqslant ck \log k$
- **Passo di induzione**: Dimostriamo la disequazione per $T(n)$:

$$\begin{aligned}
T(n) &= 2T(\lfloor \tfrac{n}{2} \rfloor) + n && \text{sost. ip. ind.} \\
&\leqslant 2c\lfloor \tfrac{n}{2} \rfloor \log \lfloor \tfrac{n}{2} \rfloor + n && \text{rimuovo l'intero inferiore} \\
&\leqslant 2c\tfrac{n}{2} \log \tfrac{n}{2} + n && \text{semplifico} \\
&\leqslant cn \log \tfrac{n}{2} + n && \log \tfrac{n}{2} = \log n - \log_2 2 = \log n - 1 \\
&\leqslant cn(\log n - 1) + n && \text{moltiplico} \\
&\leqslant cn \log n - cn + n && \text{obiettivo} \\
&\leqslant cn \log n - cn + n \overset{?}{\leqslant} cn \log n \\
&\Leftrightarrow -cn + n \leqslant 0 && \text{semplifico} \\
&\Leftrightarrow c \geqslant 1
\end{aligned}$$

- **caso base**: dimostriamo la disequazione per $T(1)$

$$T(1) = 1 \overset{?}{\leqslant} 1 \cdot c \log 1 = 0 \Rightarrow 1 \not\leqslant 0$$

É falso, ma non è un problema, non a caso si chiama notazione asintotica. Il valore iniziale di m lo possiamo scegliere noi.

- **caso base**: dimostriamo la disequazione per $T(2)$, $T(3)$

$$T(2) = 2T(\lfloor\frac{2}{2}\rfloor) + 2 = 4 \leqslant 1 \cdot c \cdot 2\log 2 \Leftrightarrow c \geqslant 2$$
$$T(3) = 2T(\lfloor\frac{3}{2}\rfloor) + 2 = 5 \leqslant 1 \cdot c \cdot 3\log 3 \Leftrightarrow c \geqslant \frac{5}{3\log 3}$$
$$T(4) = 2T(\lfloor\frac{4}{2}\rfloor) + 2 = 2T(\lfloor 2\rfloor) + 4$$

  Non è necessario provare la terza disequazione, in quanto viene espressa in base ai casi base diversi da $T(1)$ che sono già stati dimostrati e quindi possono costituire la base per la nostra induzione.

Riassumendo:

Abbiamo provato che $T(n) \leqslant cn\log n$

- Nel passo induttivo: $\forall c \geqslant 1$
- Nel caso base: $\forall c \geqslant 2, c \geqslant \frac{5}{3\log 3}$

Visto che sono tutte disequazioni con il segno $\geqslant$, è sufficiente utilizzare $c \geqslant \max\left\{1, 2, \frac{5}{3\log 3}\right\}$

Questo vale per $n = 2$, $n = 3$, e per tutti i valori di n successivi, quindi $m = 2$.

Ultimo esercizio

$$T(n) = \begin{cases} 9T(\lfloor n/3\rfloor) + n & n > 1 \\ 1 & n \leqslant 1 \end{cases}$$

Effettuiamo un tentantivo $\boxed{T(n) = \mathcal{O}(n^2)}$

Dobbiamo dimostrare che $\exists c > 0, \exists m \geqslant 0 : T(n) \leqslant cn^2, \forall n \geqslant m$

- **ipotesi induttiva**: $\exists c > 0 : T(k) \geqslant ck^2, \forall k > n$
- **passo induttivo** dimostriamo la disequazione per $T(n)$:

$$\begin{aligned} T(n) &= 9T\left(\lfloor\tfrac{n}{3}\rfloor\right) + n && \text{sost. ip. ind. con } k = \lfloor\tfrac{n}{3}\rfloor \\ &\leqslant 9c\left(\lfloor\tfrac{n}{3}\rfloor\right)^2 + n && \text{rimuovo l'intero inferiore} \\ &\leqslant 9c\left(\tfrac{n^2}{9}\right) + n && \text{semplifico il 9} \\ &= cn^2 + n && \text{obiettivo} \\ &= cn^2 + n \leqslant cn^2 \end{aligned}$$

  L'ultima disequazione risulta false per un termine di ordine inferiore.

- **ipotesi induttiva più stretta** $\exists c > 0 : T(k) \geqslant c(k^2 - k), \forall k < n$

- **passo induttivo** dimostriamo la disequazione per $T(n)$:

$$
\begin{aligned}
T(n) &= 9T\left(\lfloor \tfrac{n}{3} \rfloor\right) + n && \text{sost. ip. ind. con } k = \lfloor \tfrac{n}{3} \rfloor \\
&\leqslant 9c\left(\lfloor \tfrac{n}{3} \rfloor^2 - \lfloor \tfrac{n}{3} \rfloor\right) + n && \text{rimuovo l'intero inferiore} \\
&\leqslant 9c\left(\left(\tfrac{n}{3}\right)^2 - \tfrac{n}{3}\right) + n && \text{svolgo la potenza} \\
&\leqslant 9c\left(\tfrac{n^2}{9} - \tfrac{n}{3}\right) + n && \text{moltiplico} \\
&\leqslant cn^2 - 3cn + n && \text{obiettivo} \\
&\leqslant cn^2 - 3cn + n \overset{?}{\leqslant} cn^2 && \\
&\Leftrightarrow 2cn \geqslant cn && \text{semplifico} \\
&\Leftrightarrow c \geqslant \tfrac{1}{2} &&
\end{aligned}
$$

- **caso base**

$$
\begin{aligned}
&T(1) = 1 \leqslant c(1^2 - 1) = 0, falso \\
&T(2) = 9T(0) + 2 = 11 \leqslant c(2^2 - 2) \Leftrightarrow c \geqslant {}^{11}/_{2} \\
&T(3) = 9T(1) + 3 = 12 \leqslant c(3^2 - 3) \Leftrightarrow c \geqslant {}^{12}/_{6} \\
&T(4) = 9T(1) + 4 = 13 \leqslant c(4^2 - 4) \Leftrightarrow c \geqslant {}^{13}/_{12} \\
&T(5) = 9T(1) + 5 = 14 \leqslant c(5^2 - 5) \Leftrightarrow c \geqslant {}^{14}/_{20} \\
&T(6) = 9T(2) + 6
\end{aligned}
$$

  Non è necessario andare oltre poiché T(6) dipende da T(2) che è già stato dimostrato.

  Riassumendo i parametri scelti sono:

  - $c \geqslant \max\left\{\frac{1}{2}, \frac{11}{2}, \frac{12}{6}, \frac{13}{12}, \frac{14}{20}\right\}$
  - $m = 1$

  Nota che l'esempio combina le due difficoltà insieme, ma è stato creato artificiosamente: infatti se avessimo scelto come ipotesi più stretta $T(n) \leqslant cn^2 - bn$, il problema sui casi base non si sarebbe posto.

  Abbiamo quindi dimostrato che $T(n) \leqslant c(n^2 - n) \leqslant cn^2, \forall n \geqslant 0$, ossia che $T(n) = \Omega(n^2)$

Riassumendo

Il metodo di sostituzione è composto da tre parti:

1. si *indovina* una possibile soluzione e si formula un'ipotesi induttiva;
2. si *sostituisce* nella ricorrenza le espressioni $T(\cdot)$, utilizzando l'ipotesi induttiva;
3. si *dimostra* che la soluzione è valida anche per il caso base.

Bisogna fare attenzione a:

- ad ipotizzare soluzioni troppo "strette"
- ad alcuni casi particolari che richiedono astuzie matematiche
- ai casi base in cui compare il logaritmo in quanto potrebbe complicare le cose

0.4 Metodo dell'esperto (o delle ricorrenze comuni)

Esiste un'ampia classe di ricorrenze che possono essere risolte facilmente facendo ricorso ad alcuni teoremi, ognuno dei quali si occupa di una classe particolare di equazioni di ricorrenza.

Teorema (Ricorrenze lineari con partizione bilanciata)**.** *Siano a e b costanti intere tale che $a \geqslant 1$ e $b \geqslant 2$, e c, β costanti reali tali che $c > 0$ e $\beta \geqslant 0$. Sia $T(n)$ data dalla relazione di ricorrenza:*

$$T(n) = \begin{cases} aT\left(\frac{n}{b}\right) + cn^{\beta} & n > 1 \\ d & n \leqslant 1 \end{cases}$$

Posto $\alpha = \frac{\log a}{\log b} = \log_b a$, allora:

$$T(n) = \begin{cases} \Theta(n^{\alpha}) & \alpha > \beta \\ \Theta(n^{\alpha} \log n) & \alpha = \beta \\ \Theta(n^{\beta}) & \alpha < \beta \end{cases}$$

Commento Affrontiamo le equazioni di ricorrenza in cui la dimensione viene divisa in b parti, dove b dev'essere almeno pari a 2; algoritmo ricorsivo dev'essere richiamato a volte, dove a è almeno 1. Nella versione estesa, che vedremo fra poco, vedremo che i parametri a e b verranno "rilassati".

Prendiamo ad esempio il primo esercizio $T(n) = 4T(\frac{n}{2})$ e vediamo che si può risolvere semplicemente calcolando $\alpha = \log_b a = \log_2 4 = 2 > \beta = 1$, possiamo quindi concludere che $T(n) = \Theta(n^{\alpha}) = \Theta(n^2)$.

Dimostrazione del teorema delle ricorrenze lineari con partizione bilanciata. Assumiamo che n sia una potenza intera di b, ossia che $n = b^k$, $k = \log_b n$ poiché ci permetterà di semplificare i calcoli successivi ed è ininfluente sul risultato. Ad esempio supponiamo che l'input abbiamo dimensione $b^k + 1$ ($2^8 + 1 = 257$ bit), se estendiamo l'input fino ad una dimensione b^{k+1} ($2^8 + 1 = 512$ bit, facendo del (*padding*), l'input sarebbe stato esteso al massimo di un fattore costante b (2 nel nostro caso), il che è ininfluente al fine della complessità computazionale.

Calcoliamo l'albero delle ricorrenze per la seguente equazione di ricorrenza:

$$T(n) = \begin{cases} aT\left(\frac{n}{b}\right) + cn^{\beta} & n > 1 \\ d & n \leqslant 1 \end{cases}$$

livello	dim.	costo chiam.	no. chiamate	costo livello
0	b^k	$c\,b^{k\beta}$	1	$cb^{k\beta}$
1	b^{k-1}	$c\,b^{(k-1)\beta}$	a	$a \cdot c\,b^{k\beta}$
2	b^{k-2}	$c\,b^{(k-2)\beta}$	a^2	$a^2 \cdot c\,b^{k\beta}$
...	...	...	...	...
i	b^{k-i}	$c\,b^{(k-i)\beta}$	a^i	$a^i \cdot c\,b^{(k-i)\beta}$
...	...	...	...	...
$k-1$	b	$cb^{(k-(k-1))\beta} = c\,b^{\beta}$	a^{k-1}	$a^{k-1} \cdot c\,b^{\beta}$
k	1	d	a^k	$a^k \cdot d$

Sommando i costi totali di tutti i livelli, si ottiene:

$$
\begin{aligned}
T(n) &= da^k + \sum_{i=0}^{k-1} a^i \cdot cb^{(k-i)\beta} && \text{moltiplico} \\
&= da^k + \sum_{i=0}^{k-1} a^i \cdot cb^{k\beta} + b^{-i\beta} && \text{porto fuori i termini non dipendenti da } i \\
&= da^k + cb^{k\beta} \sum_{i=0}^{k-1} \frac{a^i}{b^{i\beta}} && \text{raccolgo } i \\
&= da^k + cb^{k\beta} \sum_{i=0}^{k-1} \left(\frac{a}{b^\beta}\right)^i
\end{aligned}
$$

A questo punto ho una formula chiusa ma non ancora nella sua forma definitiva.
Facciamo alcune osservazioni.

- $a^k = a^{\log_b n} = a^{\frac{\log n}{\log b}} = 2^{\log_2 a \frac{\log n}{\log b}} = 2^{\log_2 n \frac{\log a}{\log b}} = n^{\frac{\log a}{\log b}} = n^\alpha$
- $\alpha = \frac{\log a}{\log b} \Leftrightarrow \alpha \log b = \log a \Leftrightarrow \log b^\alpha = \log a \Leftrightarrow a = b^\alpha$
- poniamo $q = \frac{a}{b^\beta} = \frac{b^\alpha}{b^\beta} = b^{\alpha-\beta}$

Grazie alle osservazioni appena fatto possiamo sostituire i parametri nell'equazione finale:

$$
\begin{aligned}
T(n) &= da^k + cb^{k\beta} \sum_{i=0}^{k-1} \left(\frac{a}{b^\beta}\right)^i && \text{sostituisco} \\
&= dn^\alpha + cb^{k\beta} \sum_{i=0}^{k-1} q^i
\end{aligned}
$$

Da qui si aprono tre possibilità, ossia che 1. $\alpha > \beta$ 2. $\alpha = \beta$ 3. $\alpha < \beta$ Studiamoli uno per uno.

1. Caso $\boxed{\alpha > \beta}$, ne segue che $q = b^{\alpha-\beta} > 1$

$$
\begin{aligned}
T(n) &= dn^\alpha + cb^{k\beta} \sum_{i=0}^{k-1} q^i && \text{serie geometrica finita} \\
&= n^\alpha d + cb^{k\beta} \left[\frac{q^k - 1}{q - 1}\right] && \text{introduco la disequazione} \\
&\leqslant n^\alpha d + cb^{k\beta} \frac{q^k}{q-1} && \text{sostituisco: } q = \tfrac{a}{b^\beta} \Rightarrow q^k = \tfrac{a^k}{b^{k\beta}} \\
&= n^\alpha d + \frac{cb^{k\beta} a^k}{b^{k\beta}} \frac{1}{q-1} && \text{semplifico: } b^{k\beta} \\
&= n^\alpha d + \frac{ca^k}{q-1} && \text{sostituisco: } a^k = n^\alpha \\
&= n^\alpha d + \frac{cn^\alpha}{q-1} && \text{raccolgo per } n^\alpha \\
&= n^\alpha \left[d + \frac{c}{q-1}\right]
\end{aligned}
$$

Visto che d, c e q sono tutti termini positivi e costanti possiamo concludere che n^α limita superiormente l'espressione e che quindi $T(n) = \mathcal{O}(n^\alpha)$. Infine per via della componente non ricorsiva dn^α, $T(n)$ è anche $\Omega(n^\alpha)$, possiamo concludere che $T(n) = \Theta(n^\alpha)$

2. Caso $\boxed{\alpha = \beta}$, ne segue che $q = b^{\alpha-\beta} = 1$

$$\begin{aligned} T(n) &= dn^\alpha + cb^{k\beta}\sum_{i=0}^{k-1} q^i && q^i = 1^i = 1 \\ &= n^\alpha d + cn^\beta k && 1+2+\cdots+k-1 = k \\ &= n^\alpha d + cn^\alpha k && \text{sostituisco: } \beta = \alpha \\ &= n^\alpha (d + ck) && \text{raccolgo per } n^\alpha \\ &= n^\alpha (d + c\tfrac{\log n}{\log b}) && \text{sostituisco: } k = \log_b n \end{aligned}$$

Visto che dc e $\log b$ sono tutti termini positivi e costanti e che non abbiamo introdotto disequazioni, possiamo affermare che $T(n) = \Theta(n^\alpha \log n)$

3. Caso $\boxed{\alpha < \beta}$, ne segue che $q = b^{\alpha-\beta} < 1$

$$\begin{aligned} T(n) &= dn^\alpha + cb^k \sum_{i=0}^{k-1} q^i && \text{serie geometrica finita} \\ &= dn^\alpha + cb^k \left[\frac{q^k - 1}{q-1}\right] && \text{inversione, } 1-q > 0 \\ &= dn^\alpha + cb^k \left[\frac{1-q^k}{1-q}\right] && \text{introduco} \\ &\leqslant dn^\alpha + cb^k \left[\frac{1}{1-q}\right] && \text{la disequazione} \\ &= n^\alpha d + \frac{cn^\beta}{1-q} && \text{sostituisco: } b^k = n \end{aligned}$$

$n^\alpha < n^\beta$ quindi considero il polinomio di grado maggiore, di conseguenza $T(n)$ è $\mathcal{O}(n^\beta)$. Poiché $T(N) = \Omega(n^\beta)$ per via del termine non ricorsivo, possiamo affermare che $T(n) = \Theta(n^\beta)$.

Fine dimostrazione. □

Teorema (Ricorrenze lineari con partizione bilanciata estesa)**.** *Sia $a \geqslant 1$, $b > 1$, $f(n)$ asintoticamente positiva, e sia*

$$T(n) = \begin{cases} aT\left(\frac{n}{b}\right) + f(n) & n > 1 \\ d & n \leqslant 1 \end{cases}$$

Sono dati tre casi:

1. $\exists \varepsilon > 0 : f(n) = \mathcal{O}(n^{\alpha-\varepsilon})$ *allora* $T(n) = \Theta(n^\alpha)$;
2. $f(n) = \Theta(n^\alpha)$ *allora* $T(n) = \Theta(f(n) \log n)$;
3. $\exists \varepsilon > 0 : f(n) = \mathcal{O}(n^{\alpha+\varepsilon}) \wedge$
   $\exists c : 0 < c < 1, \exists m > 0 :$
   $af(\frac{n}{b}) \leqslant cf(n), \forall n \geqslant m$ *allora* $T(n) = \Theta(f(n))$.

Non vedremo la dimostrazione poiché è troppo complessa.

Commento Il teorema precendente funzionava con n^β, aveva una serie di condizioni semplificative, ora non ci sono più. Nel secondo caso se $f(n) = n^\beta$ ritorniamo esattamente al secondo caso del teorema precendente, ma qui possiamo prendere in cnsiderazione funzioni più complesse.

Esercizi

- $T(n) = 9T(\frac{n}{3}) + n$
- $T(n) = T(\frac{2}{3}n) + n$
- $T(n) = 3T(\frac{n}{4}) + n \log n$
- $T(n) = 2T(\frac{n}{2}) + n \log n$

Teorema (Ricorrenze lineari di ordine costante)**.** *Siano* $\{a_1, a_2, \ldots, a_n\}$ *costanti intere non negative, con* h *costante positiva,* c *e* β *costanti reali tali che* $c > 0$ *e* $\beta \geqslant 0$*, e sia* $T(n)$ *definita dalla relazione di ricorrenza:*

$$T(n) = \begin{cases} \sum_{1 \leqslant i \leqslant h} a_i T(n-1) + cn^{\beta} & n > m \\ \Theta(1) & n \leqslant m \leqslant h \end{cases}$$

Posto $a = \sum_{1 \leqslant i \leqslant h} a_i$*, allora:*

1. $T(n)$ *è* $\Theta(n^{\beta+1})$*, se* $a = 1$*;*
2. $T(n)$ *è* $\Theta(a^n n^{\beta})$*, se* $a \geqslant 2$*.*

Commento Questo teorema tratta ricorrenze lineari *di ordine costante* perché tutte le volte rimuoviamo dalla dimensione di input n una quantità costante.

Esercizi

- $T(n) = T(n-10) + n^2$
- $T(n) = T(n-2) - T(n-1) + 1$