2 Analisi della complessità

$$\boxed{f(n) = 10n^3 + 2n^2 + 7 \stackrel{?}{=} \mathcal{O}(n^3)}$$

Limite superiore: Dobbiamo dimostrare che $\exists c > 0, \exists m \geqslant 0 : f(n) \leqslant cn^3, \forall n \geqslant m$

$$\begin{aligned} f(n) &= 10n^3 + 2n^2 + 7 \\ &\leqslant 10n^3 + 2n^3 + 7 && \forall n \geqslant 1 \\ &\leqslant 10n^3 + 2n^3 + 7n^3 \\ &= 19n^3 && \textit{sommiamo i termini} \\ &\stackrel{?}{\leqslant} cn^3 && \textit{esiste una certa costante } c \textit{ per la quale } f(n) \leqslant cn^3 \ ? \end{aligned}$$

che è vera per ogni $c \geqslant 19$ e per ogni $n \geqslant 1$, quindi $m = 1$.

$$\boxed{f(n) = 3n^2 + 2n^2 + 7n \stackrel{?}{=} \Theta(n^2)}$$

Limite inferiore: Dobbiamo dimostrare che
$\exists c_1 > 0, \exists m_1 \geqslant 0 : f(n) \geqslant c_1 n^2, \forall n \geqslant m_1$

$$\begin{aligned} f(n) &= 3n^2 + 2n^2 + 7n \\ &\geqslant 3n^2 && \forall n \geqslant 0 \\ &\stackrel{?}{\geqslant} c_1 n^2 \end{aligned}$$

che è vera per ogni $c_1 \leqslant 3$ e per ogni $n \geqslant 0$,
quindi $m_1 = 0$.
Ho dimostrato quindi che $f(n) = \Omega(n^2)$
Limite superiore: Dobbiamo dimostrare che
$\exists c_2 \geqslant 0, \exists m_2 \geqslant 0 : f(n) \leqslant c_2 n^2, \forall n \geqslant m_2$

$$\begin{aligned} f(n) &= 3n^2 + 2n^2 + 7n \\ &\leqslant 3n^2 + 7n^2 && \forall n \geqslant 1 \\ &\leqslant 10n^2 \\ &\stackrel{?}{\leqslant} c_2 n^2 \end{aligned}$$

che è vera per ogni $c_2 \geqslant 10$ e per ogni $n \geqslant 1$,
quindi $m_2 = 1$.
Ho dimostrato quindi che $f(n) = \mathcal{O}(n^2)$

Notazione Θ: $\exists c_1 > 0, \exists c_2 > 0, \exists m \geqslant 0 : c_1 n^2 \leqslant f(n) \leqslant c_2 n^2, \forall n \geqslant m$
Con questi paramentri:

- $c_1 = 3$
- $c_1 = 10$
- $m = max\{m_1, m_2\} = max\{0, 1\} = 1$ un valore dopo il quale la nostra proprietà è provata

Ho dimostrato quindi che $f(n) = \Theta(n^2)$

2.1 Complessità dei problemi

Definizione 2.1 ($\mathcal{O}(f(n))$ — Limite superiore)**.** *Un problema ha complessità* $\mathcal{O}(f(n))$ se esiste almeno un algoritmo *che ha complessità* $\mathcal{O}(f(n))$.

Definizione 2.2 ($\Omega(f(n))$ — Limite inferiore)**.** *Un problema ha complessità* $\Omega(f(n))$ se tutti i possibili algoritmi *che lo risolvono hanno complessità* $\Omega(f(n))$.

2.2 Equazioni di ricorrenza

2.2.1 Moltiplicare numeri complessi

2.2.2 Moltiplicare numeri binari

Moltiplicazione di Karatsuba

$$\begin{aligned} A_1 &= a \times c \\ A_2 &= b \times d \\ m &= (a+b) \times (c+d) = ac + ad + bc + bd \\ A_3 &= m - A_1 - A_2 = ad + bc \end{aligned}$$

calcolo i valori intermedi

evito una moltiplicazione

$$T(n) = \begin{cases} c_1 & \text{se } n = 1 \\ 3T(n/2) + c_2 \cdot n & \text{se } n > 1 \end{cases} = \mathcal{O}(n^{1.58\ldots})$$

2.3 Algoritmi di ordinamento

2.3.1 Selection sort

2.3.2 Insertion sort

2.3.3 Merge sort

Quando un'array diventa vuoto, copia tutti i valori dall'array rimanente nell'array ordinato.

2.4 Analisi di algoritmi

2.4.1 Proprietà della notazione asintotica

Definizione 2.3 (Dualità)**.**

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

La dimostrazione avviene tramite passaggi algebrici:

$$\begin{aligned} f(n) = \mathcal{O}(g(n)) &\Leftrightarrow f(n) \leqslant \boldsymbol{c}\ g(n), \forall n \geqslant m \\ &\Leftrightarrow g(n) \leqslant \tfrac{\mathbf{1}}{\boldsymbol{c}}\ f(n), \forall n \geqslant m \\ &\Leftrightarrow g(n) = \boldsymbol{c'}\ f(n) \\ &\Leftrightarrow g(n) = \Omega(f(n)) \end{aligned}$$

ribalto la disequazione e divido per $\frac{1}{c}$

rinomino $\frac{1}{c}$ a c'

passo alla notazione dei quantificatori

Definizione 2.4 (Sommatoria (sequenza di algoritmi))**.**

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

Ne manca una.

Definizione 2.5 (Simmetria)**.**

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

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

Definizione 2.6 (Transitività)**.**

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

La dimostrazione è banale:

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

2.5 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:*

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

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

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

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

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

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. $\exists \varepsilon > 0 : f(n) = \mathcal{O}(n^{\alpha + \varepsilon}) \land \exists c : 0 < c < 1, \exists m > 0 : af\left(n/b\right) \leqslant cf(n), \forall n \geqslant m$ *allora* $T(n) = \Theta(f(n))$.

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:*

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

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* $a \geqslant 2$.