Esercizio 1. Dimostrare le seguenti affermazioni usando le definizioni di $\Theta(\cdot)$ e $O(\cdot)$. In questo esercizio, così come nei seguenti, log denota il logaritmo in base 2.

1. $(n^n + n2^n + 5^n)(n! + 5^n) = O(n^n n!)$
2. $(n! + 2^n)(n^3 + \log(n^2 + 1)) = O(n^3 n!)$ $\rightsquigarrow$ simile a 1.
3. $n! \neq O(2^n)$
4. $\log n! = \Theta(n \log n)$

1) $\overbrace{(n^n + n2^n + 5^n)(n! + 5^n)}^{f_1(n)} = O(n^n \cdot n!) \Leftrightarrow \exists c, n_0 > 0 : 0 < f_1(n) \leq c n^n n! \ \forall n \geq n_0$

$n^n n! + n2^n n! + 5^n n! + n^n 5^n + n2^n 5^n + 5^n \cdot 5^n$

IDEA: provo $c = 1$ per trovare n_0 che soddisfa la disuguaglianza

- $n^n n! = O(n^n n!)$
- $n 2^n \cancel{n!} \leq n^n \cancel{n!}$

$\Leftrightarrow n 2^n \leq n^n$

$\Leftrightarrow 2^n \leq \frac{n^n}{n} = n^{n-1}$

$\Leftrightarrow 1 \leq \frac{n^{n-1}}{2^n} = \frac{\overbrace{n \cdot n \cdot \ldots \cdot n}^{n-1 \text{ volte}}}{2 \cdot 2 \cdot \ldots \cdot 2} \cdot \frac{1}{2} = \frac{n}{2} \cdot \frac{n}{2} \cdot \ldots \cdot \frac{n}{2} \cdot \frac{1}{2} = \left(\frac{n}{2}\right)^{n-1} \cdot \frac{1}{2}$

$\Leftrightarrow \left(\frac{n}{2}\right)^{n-1} \geq 2$

$\Leftrightarrow n \geq 3$

$\rightarrow$ prendo $n_0 = 3$ così $\frac{n_0}{2} \geq 2 \Rightarrow \left(\frac{n_0}{2}\right)^{n-1} \geq 2 \ \forall n \geq n_0 \Rightarrow n 2^n n! = O(n^n n!)$

- $5^n \cdot n! \leq n^n n!$

$\Leftrightarrow 5^n \leq n^n$

$\Leftrightarrow 1 \leq \frac{n^n}{5^n} = \frac{n \cdot n \cdot \ldots \cdot n}{5 \cdot 5 \cdot \ldots \cdot 5} = \overbrace{\frac{n}{5} \cdot \ldots \cdot \frac{n}{5}}^{n \text{ volte}} = \left(\frac{n}{5}\right)^n$

$\Leftrightarrow \frac{n}{5} \geq 1 \Leftrightarrow n \geq 5$

$\rightsquigarrow$ $n_0 = 5$

- $\cancel{n^n} 5^n \leq \cancel{n^n} n!$

$\Leftrightarrow \quad 5^n \leq n!$

spezzo qui

$\Leftrightarrow \quad 1 \leq \frac{n!}{5^n} = \frac{n\cdot(n-1)\cdot(n-2)\cdot \_\_ \cdot 6\cdot 5\cdot 4\cdot 3\cdot 2\cdot 1}{5\cdot 5\cdot 5\cdot \_\_ \cdot 5\cdot 5\cdot 5\cdot 5\cdot 5} = \overset{\geq 1}{\boxed{\frac{n\cdot(n-1)\cdot \_\_ \cdot 5}{5\cdot 5\cdot \_\_ \cdot 5}}} \cdot \overset{< 1}{\boxed{\frac{4!}{5^4}}}$

$\Leftrightarrow \quad 1 \leq \frac{n\cdot(n-1)\cdot \_\_ \cdot 5}{5\cdot 5\cdot \_\_ \cdot 5} \cdot \frac{1}{\alpha} \qquad \left[\frac{1}{\alpha} = \frac{4!}{5^4}\right]$

$\Leftrightarrow \quad 1 \leq \frac{n}{5}\cdot \underbrace{\frac{n-1}{5}}_{\geq 1}\cdot \underbrace{\_\_}_{\geq 1}\cdot \underbrace{\frac{5}{5}}_{\geq 1}\cdot \frac{1}{\alpha}$

- E' sufficiente che

$1 \leq \frac{n}{5}\cdot\frac{1}{\alpha} \qquad \left[\leq \frac{n}{5}\cdot\frac{n-1}{5}\cdot \_\_ \cdot \frac{5}{5}\cdot\frac{1}{\alpha}\right]$

$\Leftrightarrow \quad n \geq 5\alpha = \frac{5^5}{4!}$

$\rightsquigarrow$ prendo $n_0 = \left\lceil \frac{5^5}{4!} \right\rceil$

Esempio

$n\cdot(n-1)\cdot \_\_ \cdot 2\cdot 1 \leq n\cdot n\cdot(n-2)\cdot(n-3)\cdot \_\_ \cdot 2\cdot 1$

$\leq n\cdot n\cdot n\cdot n\cdot \_\_ \cdot n\cdot n$

$= n^n \qquad \forall n \geq 1$

$5! = 5\cdot 4\cdot 3\cdot 2\cdot 1 \leq 5\cdot 5\cdot 3\cdot 2\cdot 1 \leq 5\cdot 5\cdot 5\cdot 2\cdot 1$

$\leq 5\cdot 5\cdot 5\cdot 5\cdot 5 \leq 5^5$

- $n\, 2^n 5^n \le n^n n!$

$\Leftrightarrow \quad 2^n 5^n \le n^{n-1} \cdot n!$ $\qquad [\forall n \geq 1]$

$\Leftrightarrow \quad 1 \le \dfrac{n^{n-1} \cdot n!}{10^n} = \dfrac{n^{n-1} \cdot n!}{10^n}$

$\Leftrightarrow \quad 1 \le \dfrac{n}{10} \cdot \dfrac{n-1}{10} \cdot \_\_ \cdot \dfrac{10}{10} \cdot \dfrac{9!}{10^9} \cdot \overbrace{n \cdot n \cdot \_\_ \cdot n}^{n-1}$

- E' sufficiente che

$1 \le \dfrac{n}{10} \cdot \dfrac{9!}{10^9}$ $\qquad \left[\le \dfrac{n \cdot (n-1) \cdot \_\_ \cdot 2 \cdot 1}{10^n} \cdot \overbrace{n \cdot n \cdot \_\_ \cdot n}^{n-1} \right]$

$\Leftrightarrow \quad n \geq \dfrac{10^{10}}{9!}$

$\rightarrow$ prendo $n_0 = \left\lceil \dfrac{10^{10}}{9!} \right\rceil$

- $5^n \cdot 5^n \le n^n \cdot n!$

$\Leftrightarrow \quad 1 \le \dfrac{n^n \cdot n!}{5^n \cdot 5^n} = \dfrac{n^n}{5^n} \cdot \dfrac{n!}{5^n} \le \dfrac{n^n}{5^n} \cdot \dfrac{n^n}{5^n} = \left(\dfrac{n}{5}\right)^n \cdot \left(\dfrac{n}{5}\right)^n = \left(\dfrac{n}{5}\right)^{2n}$

$\Leftrightarrow \quad 1 \le \left(\dfrac{n}{5}\right)^{2n}$

$\Leftrightarrow \quad n \geq 5$

3) $n! \neq O(2^n)$

- per assurdo supp. $n! = O(2^n) \iff \exists c, n_0 > 0 : 0 \leq n! \leq c\, 2^n \quad \forall n \geq n_0$
- quindi $\exists c, n_0 > 0$ tali per cui

$$n! \leq c\, 2^n \iff \frac{n!}{2^n} \leq c \qquad \forall n \geq n_0$$

ossia la successione $\frac{n!}{2^n}$ sarebbe superiormente limitata da c da n_0 in poi

- questo e' assurdo, infatti

$$\frac{n!}{2^n} = \frac{n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1}{2 \cdot 2 \cdot \ldots \cdot 2 \cdot 2} = \frac{n}{2} \cdot \frac{n-1}{2} \cdot \ldots \cdot \frac{1}{2}$$

$$\geq \frac{n}{2} \cdot \frac{1}{2} = \frac{n}{4} \qquad \forall n \in \mathbb{N} \qquad (1)$$

- inoltre

$$\lim_{n \to +\infty} \frac{n}{4} = +\infty \qquad (2)$$

TEO. CONFRONTO (1), (2)

$$\implies \lim_{n \to +\infty} \frac{n!}{2^n} = +\infty \iff \forall c > 0 \quad \exists n_0 \in \mathbb{N} : \frac{n!}{2^n} > c \quad \forall n \geq n_0 \qquad \square$$

4) $\log(n!) = \Theta(n \log(n))$

- $\log(n!) = \log(n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1)$

$$= \log(n) + \log(n-1) + \ldots + \log(2) + \log(1) \qquad \forall n \geq 1$$

$$\leq \log(n) + \log(n) + \ldots + \log(n) + \log(n)$$

$$= n \log(n)$$

$$= O(n \log n)$$

- $\log(n!) = \log(n) + \log(n-1) + \ldots + \log\left(\frac{n}{2}+1\right) + \log\left(\frac{n}{2}\right) + \ldots + \log(1)$

$$\geq \log\left(\frac{n}{2}\right) + \log\left(\frac{n}{2}-1\right) + \ldots + \log(2) + \log(1)$$

$$\geq \log\left(\frac{n}{2}\right) + \log\left(\frac{n}{2}\right) + \ldots + \log\left(\frac{n}{2}\right) + \log\left(\frac{n}{2}\right) \qquad \forall n \geq 1$$

$$\geq \frac{n}{2} \log\left(\frac{n}{2}\right) = \frac{n}{2}\left[\log(n) - \log(2)\right] = \frac{n}{2} \log(n) - \frac{n}{2}$$

$$= \Omega(n \log(n))$$

$$\Rightarrow \log(n!) = \Theta(n \log(n)).$$

Esercizio 2. Dimostrare o confutare le seguenti affermazioni:

1. $n \log n = O(\log n!)$ ✓ $\log n! = \Theta(n \log n) \xrightarrow{\Theta \text{ simm.}} n \log n = \Theta(\log n!) \ldots$
2. $\log_{16} n = O(\log_4 n)$ e $\log_8 n = O(\log_{10} n)$ ✓ cambio base log
3. $\log(n^2+1) = O(\log n)$ ✓ $\log(n^2+1) \sim \log(n^2) = 2\log(n)$
4. Per risolvere un certo problema abbiamo a disposizione due algoritmi, A_1 e A_2. Se l'input ha dimensione n, A_1 impiega esattamente $n^2 2^n$ passi, mentre A_2 impiega esattamente $n!$ passi. Al crescere di n, A_1 impiega un numero minore di passi rispetto ad A_2. ✓

4. $T_{A_1}(n) = \Theta(n^2 2^n)$, $T_{A_2}(n) = \Theta(n!)$

$$\circ \quad \lim_{n\to\infty} \frac{n^2 2^n}{n!} = \lim_{n\to\infty} \frac{n \cdot n \cdot 2 \cdot \_\_ \cdot 2}{n \cdot (n-1) \cdot \_\_ \cdot 1} \cdot 2^2$$

$$= \lim_{n\to\infty} 1 \cdot \frac{\cancel{n}}{n-1}\,{}_{\to 1} \cdot \frac{\cancel{2}}{n-2}\,{}_{\to 0} \cdot \_\_ \cdot \frac{2}{2} \cdot \frac{2}{1} \cdot 2^2$$

$$= \lim_{n\to\infty} 1 \cdot \frac{n}{n-1} \cdot \prod_{i=2}^{n-2} \underbrace{\frac{2}{n-i}}_{\text{ogni fattore tende a } \emptyset} \cdot 2^3$$

$$= 1 \cdot 1 \cdot \prod_{i=2}^{n-2} \emptyset \cdot 2^3$$

$$= \emptyset$$

quindi $n^2 2^n = o(n!) \iff \forall c>0\ \exists n_0>0 : 0 \le n^2 2^n < cn!\ \forall n \geq n_0$

Esercizio 3. Fornire un limite asintotico stretto per ognuna delle seguenti funzioni $f(n)$:

1. $f(n) = (n^2+1)^{10} = \Theta(n^{20})$
2. $f(n) = 2n\log(n+2)^2 + (n+2)^2\log\frac{n}{2}$ $\sim n\log n^2 + n^2\log n$ $= \Theta(n^2\log n)$
3. $f(n) = 2^{n+1} + 3^{n-1}$ $= \Theta(3^n)$
4. $f(n)$ è la somma dei primi n interi positivi $\rightarrow$ somma di Gauss, $\Theta(n^2)$

3. $$\lim_{n\to\infty} \frac{2^{n+1}+3^{n-1}}{3^n} = \lim_{n\to\infty} \underbrace{\frac{2^{n+1}}{3^n}}_{\to 0} + \underbrace{\frac{3^{n-1}}{3^n}}_{\to \frac{1}{3}} = 0 + \frac{1}{3} = \frac{1}{3}$$

$$\Rightarrow \quad 2^{n+1} + 3^{n-1} = \Theta(3^n)$$

Esercizio 4. Quale delle seguenti sequenze di funzioni è tale che ogni funzione è O della successiva?

1. $\log(\log n),\ n,\ \log n,\ n^n$
2. $\log n,\ 2^{2^n},\ n^n,\ n!$
3. $\log(\log n),\ \log n,\ n,\ 2^{2^n}$
4. nessuna delle precedenti

$g(n) = \Omega(f(n)) \Leftrightarrow \exists k, \tilde{n} > 0 : 0 \le k f(n) \le g(n) \quad \forall n \geq \tilde{n}$

Esercizio 5. Siano $f(n)$ e $g(n)$ funzioni positive tali che $f(n) = O(g(n))$. Dimostrare che $g(n) = \Omega(f(n))$.

$$
\begin{aligned}
f(n) = O(g(n)) &\Leftrightarrow \exists c, n_0 > 0 : 0 \le f(n) \le c\, g(n) \quad \forall n \geq n_0 \\
&\Leftrightarrow \exists c, n_0 > 0 : 0 \le \frac{1}{c} f(n) \le g(n) \quad \forall n \geq n_0 \\
&\Leftrightarrow \exists k, n_0 > 0 : 0 \le k f(n) \le g(n) \quad \forall n \geq n_0 \\
&\Leftrightarrow g(n) = \Omega(f(n)) \qquad \square
\end{aligned}
$$

Esercizio 6. Sia k un intero positivo. Dimostrare che un insieme di n elementi ha $O(n^k)$ sottoinsiemi di cardinalità k.

- n. sottoinsiemi k elementi

$$
\begin{aligned}
\binom{n}{k} := \frac{n!}{k!(n-k)!} &= \frac{n \cdot (n-1) \cdot \ldots \cdot (n-k+1)}{k!} \\
&\le \underbrace{n \cdot (n-1) \cdot \ldots \cdot (n-k+1)}_{n-(n-k+1)+1 \,=\, k \text{ elementi}} \qquad [k! > 0] \\
&\le n^k .
\end{aligned}
$$

Esercizio 7. Si consideri il seguente algoritmo, il cui input è un array $A[0..n-1]$ di interi di dimensione n. Quale problema risolve? Determinare un limite asintotico stretto per il tempo di esecuzione nel caso peggiore. Esiste un algoritmo più efficiente nel caso in cui l'array in input sia *ordinato*?

$\rightarrow \Theta(n)$

scorro l'array in una passata

Algorithm UNIQUEELEMENTS(A)

```
for i ← 0 to n − 2 do                      0 → n-2
    for j ← i + 1 to n − 1 do              i+1 → n-1
        if A[i] = A[j] then return false   } c
        end if
    end for
end for
return true
```

$$T(n) = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} c = \sum_{i=0}^{n-2} c \cdot \sum_{j=i+1}^{n-1} 1 = \sum_{i=0}^{n-2} c \cdot [(n-1)-(i+1)+1]$$

$$= \sum_{i=0}^{n-2} c \cdot (n-i-1) = c \cdot \sum_{i=0}^{n-2} (n-i-1)$$

$[k := n-i-1]$

$$= c \cdot \sum_{k=1}^{n-1} k$$

$$= c \cdot \left(\sum_{k=1}^{n} k - n \right)$$

$$= c \cdot \left(\frac{n(n+1)}{2} - n \right) = \Theta(n^2).$$

nuovi indici : basta sostituire $i=0$, $i=n-2$ nella def. di k

$i \in [0..n-2] \Rightarrow k \in [n-1..1]$

Esercizio 8. Si consideri il seguente algoritmo, il cui input è un intero positivo n. Quale problema risolve? Determinare un limite asintotico stretto per il tempo di esecuzione nel caso peggiore.

Algorithm BINARY(n)

```
count ← 1                      c1
while n > 1 do
    count ← count + 1          c2 }
    n ← ⌊n/2⌋                  c3 }  entro ⌊log n⌋ volte
end while
return count
```

- perché il corpo del while è eseguito $\lfloor \log n \rfloor$ volte ?
  - ad ogni iterazione dimezzo l'input, inizialmente è n. Quindi all'entrata del ciclo della i-esima iterazione il valore n_i sarà

$$n_i = \left\lfloor \frac{n}{2^{i-1}} \right\rfloor$$

  - entro solo quando

$$\begin{aligned} n_i > 1 \quad &\iff \left\lfloor \frac{n}{2^{i-1}} \right\rfloor > 1 \\ &\iff n > 2^{i-1} \\ &\iff \log n > \log 2^{i-1} = i-1 \\ &\iff i < \log n + 1 \\ &\iff i \le \lfloor \log n \rfloor \end{aligned}$$

$$\Rightarrow \quad T(n) = c_1 + \lfloor \log n \rfloor (c_2 + c_3) = \Theta(\log n) .$$