Algoritmica

Esercizi su relazioni di ricorrenza

G. Prencipe
giuseppe.prencipe@unipi.it

Il Master Theorem

Si applica a ricorrenze della forma

$$T(n) = a\,T(n/b) + f(n)$$

dove $a \geq 1$, $b > 1$, e f è asintoticamente positiva

$T(n) = a\,T(n/b) + f(n)$

I tre casi di applicazione

Confronta $f(n)$ con $n^{\log_b a}$:

1. $f(n) = O(n^{\log_b a - \varepsilon})$ per qualche costante $\varepsilon > 0$.
   - $f(n)$ cresce polinomialmente più lentamente di $n^{\log_b a}$ (di un fattore n^{ε})

Soluzione: $T(n) = \Theta(n^{\log_b a})$

$T(n) = a\,T(n/b) + f(n)$

I tre casi di applicazione

Confronta $f(n)$ con $n^{\log_b a}$:

2. $f(n) = \Theta(n^{\log_b a} \lg^k n)$ per qualche costante $k \geq 0$
   - $f(n)$ e $n^{\log_b a}$ crescono allo stesso modo

Soluzione: $T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$

Nota: questa è la generalizzazione di

$$f(n) = \Theta(n^{\log_b a}) \rightarrow \mathrm{T(n)} = \Theta(\mathrm{n}^{\log_b a}\, lg\mathrm{n})$$

$T(n) = a\,T(n/b) + f(n)$

I tre casi di applicazione

Confronta $f(n)$ con $n^{\log_b a}$:

3. $f(n) = \Omega(n^{\log_b a + \varepsilon})$ per qualche costante $\varepsilon > 0$
   - $f(n)$ cresce polinomialmente più veloce di $n^{\log_b a}$ (di un fattore n^{ε}), ***e***
   - $f(n)$ **soddisfa la *condizione*** che

     $a f(n/b) \leq c f(n)$ per qualche costante $c < 1$

Soluzione: $T(n) = \Theta(f(n))$.

Esercizio 1

$$T(n) = 4T(n/2) + n$$

Master Theorem....

Esercizio 1

$$T(n) = 4T(n/2) + n$$

$$a = 4, b = 2$$

Esercizio 1

$$T(n) = 4T(n/2) + n$$

$a = 4,\ b = 2 \Rightarrow n^{\log_b a} = n^2$ **e** $f(n) = n$

CASO ????

Esercizio 1

$$T(n) = 4T(n/2) + n$$

$a = 4, b = 2 \Rightarrow n^{\log_b a} = n^2$ **e** $f(n) = n$

CASO 1!

Esercizio 1

$$T(n) = 4T(n/2) + n$$

$a = 4,\ b = 2 \Rightarrow n^{\log_b a} = n^2$ **e** $f(n) = n$

CASO 1: $f(n) = O(n^{2-\varepsilon})$ per $\varepsilon = 1$

$\therefore\ T(n) = \Theta(n^2)$

Esercizio 2

$$T(n) = 4T(n/2) + n^2$$

$\boxed{\text{Master Theorem....}}$

Esercizio 2

$$\boldsymbol{T(n) = 4T(n/2) + n^2}$$

$$a = 4,\ b = 2 \Rightarrow n^{\log_b a} = n^2 \quad \mathbf{e} \quad f(n) = n^2$$

Esercizio 2

$$T(n) = 4T(n/2) + n^2$$

$a = 4, b = 2 \Longrightarrow n^{\log_b a} = n^2$ **e** $f(n) = n^2$

CASO ????

Esercizio 2

$$T(n) = 4T(n/2) + n^2$$

$a = 4, b = 2 \Rightarrow n^{\log_b a} = n^2$ **e** $f(n) = n^2$

CASE 2: $f(n) = \Theta(n^2 \lg^0 n), k = 0$

$\therefore\ T(n) = \Theta(n^2 \lg n)$

Esercizio 3

$$T(n) = 4T(n/2) + n^3$$

$\boxed{\text{Master Theorem....}}$

Esercizio 3

$$T(n) = 4T(n/2) + n^3$$

$$a = 4, b = 2 \Longrightarrow n^{\log_b a} = n^2 \quad \textbf{e} \quad f(n) = n^3$$

Esercizio 3

$$T(n) = 4T(n/2) + n^3$$

$a = 4,\ b = 2 \Rightarrow n^{\log_b a} = n^2$ **e** $f(n) = n^3$

CASO ????

Esercizio 3

$$T(n) = 4T(n/2) + n^3$$

$a = 4$, $b = 2 \Rightarrow n^{\log_b a} = n^2$ **e** $f(n) = n^3$

CASO 3: $f(n) = \Omega(n^{2+\varepsilon})$ per $\varepsilon = 1$

Esercizio 3

$$T(n) = 4T(n/2) + n^3$$

$a = 4$, $b = 2 \Longrightarrow n^{\log_b a} = n^2$ **e** $f(n) = n^3$

CASO 3: $f(n) = \Omega(n^{2+\varepsilon})$ per $\varepsilon = 1$

….basta così????

$a\,f(n/b) \leq c\,f(n)$ per qualche costante $c < 1$

Esercizio 3

$$\boldsymbol{T(n) = 4T(n/2) + n^3}$$

$a = 4,\ b = 2 \Longrightarrow n^{\log_b a} = n^2$ **e** $f(n) = n^3$

CASO 3: $f(n) = \Omega(n^{2+\varepsilon})$ per $\varepsilon = 1$

e $4(n/2)^3 \leq cn^3$ per $c = 1/2$

$\therefore\ T(n) = \Theta(n^3)$

Esercizio 4

$$T(n) = 4T(n/2) + n^2/\log n$$

Risolvere....

Esercizio 4

Proviamo con il MT....

$$T(n) = 4T(n/2) + n^2/\log n$$

$$a = 4,\ b = 2 \Rightarrow n^{\log_b a} = n^2 \quad \textbf{e} \quad f(n) = n^2/\log n$$

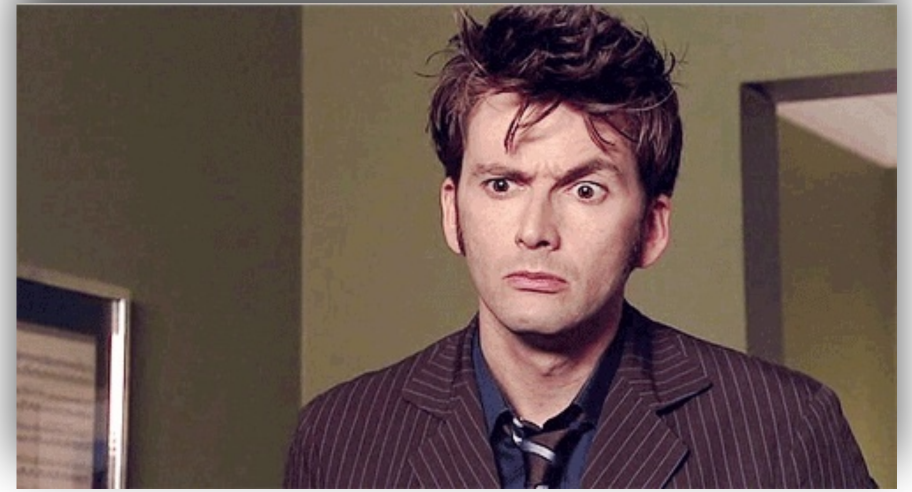
Esercizio 4

Proviamo con il MT….

$$T(n) = 4T(n/2) + n^2/\log n$$

$a = 4,\ b = 2 \Rightarrow n^{\log_b a} = n^2$ **e** $f(n) = n^2/\log n$

Si può applicare il Master Theorem????



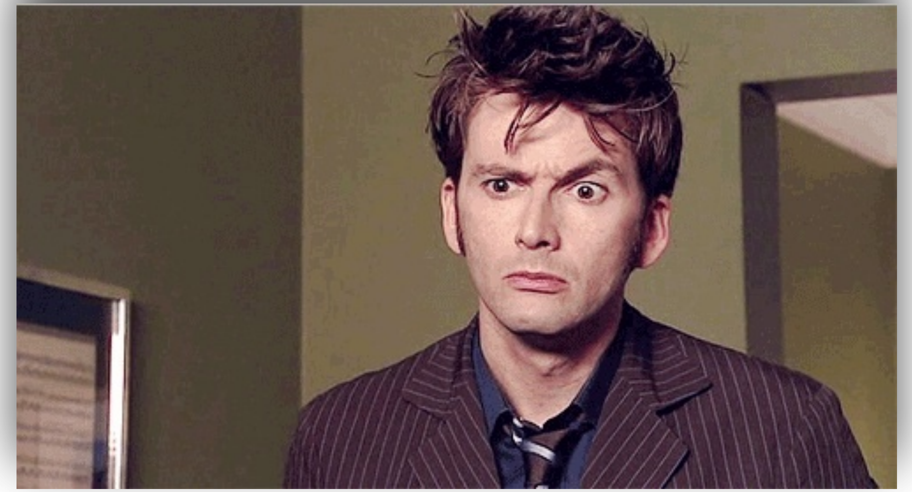
Esercizio 4

Proviamo con il MT....

$$T(n) = 4T(n/2) + n^2/\log n$$

$$a = 4,\ b = 2 \Rightarrow n^{\log_b a} = n^2 \quad \textbf{e} \quad f(n) = n^2/\log n$$

Si può applicare il Master Theorem????



$n^{\log_b a} = n^2$ vs $n^2/\log n$

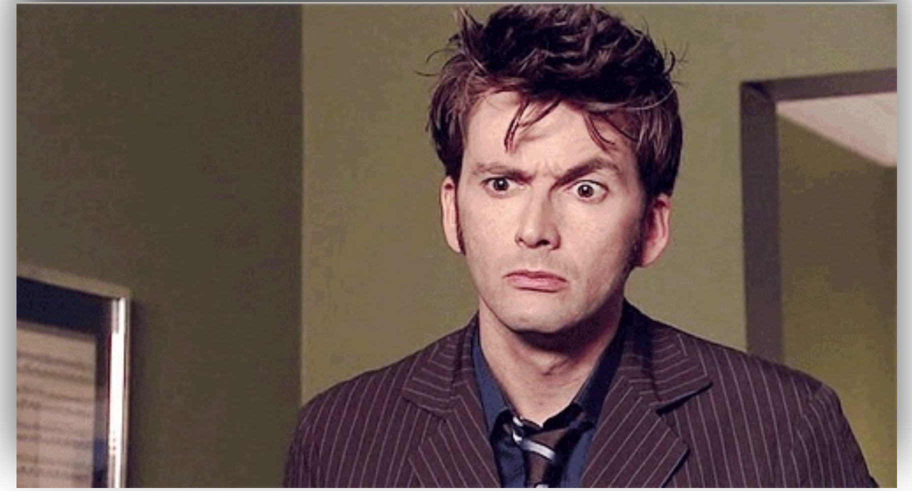
Esercizio 4

Proviamo con il MT....

$$T(n) = 4T(n/2) + n^2/\log n$$

$$a = 4, b = 2 \Rightarrow n^{\log_b a} = n^2 \quad \textbf{e} \quad f(n) = n^2/\log n$$

Si può applicare il Master Theorem????



$n^{\log_b a} = n^2$ vs $n^2/\log n$

Quali casi possiamo sicuramente escludere?

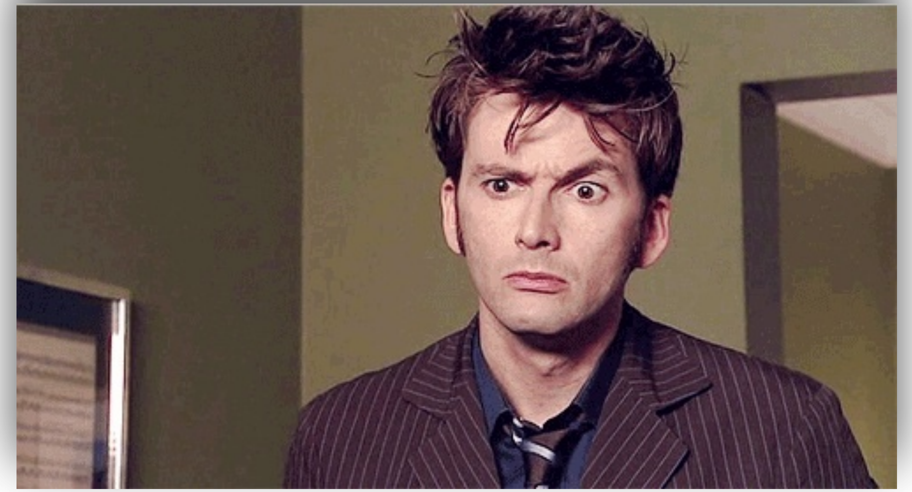
Esercizio 4

Proviamo con il MT....

$$T(n) = 4T(n/2) + n^2/\log n$$

$$a = 4,\ b = 2 \Rightarrow n^{\log_b a} = n^2 \quad \textbf{e} \quad f(n) = n^2/\log n$$

Si può applicare il Master Theorem????



$$n^{\log_b a} = n^2 \text{ vs } n^2/\log n$$

$$n^2/\log n = O(n^2)$$

Quali casi possiamo sicuramente escludere?

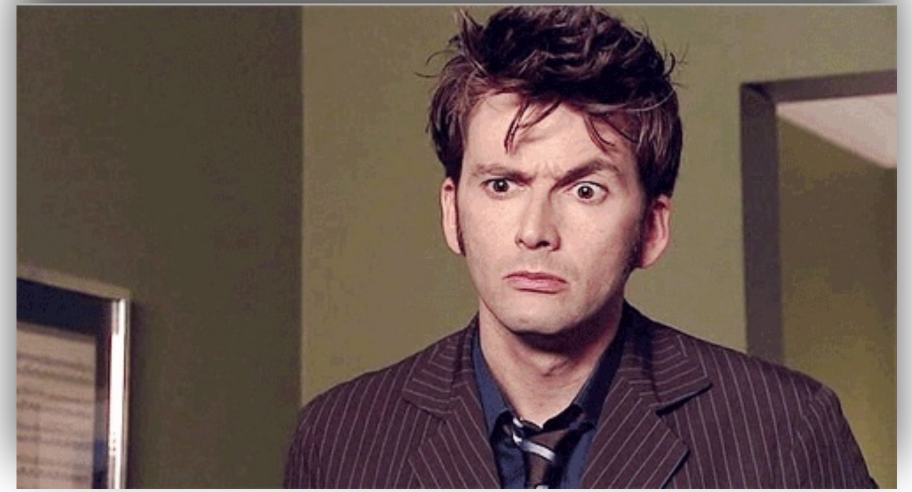
Esercizio 4

Proviamo con il MT....

$$T(n) = 4T(n/2) + n^2/\log n$$

$$a = 4,\ b = 2 \Rightarrow n^{\log_b a} = n^2 \quad \textbf{e} \quad f(n) = n^2/\log n$$

Si può applicare il Master Theorem????



$n^{\log_b a} = n^2$ vs $n^2/\log n$

$n^2/\log n = O(n^2)$

Casi 2 e 3 NO!!!!

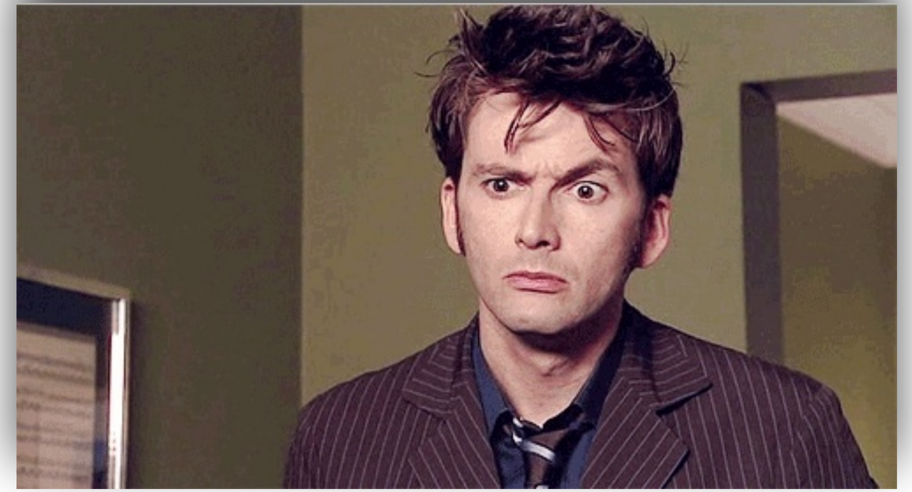
Esercizio 4

Proviamo con il MT....

$$T(n) = 4T(n/2) + n^2/\log n$$

$$a = 4, b = 2 \Rightarrow n^{\log_b a} = n^2 \quad \textbf{e} \quad f(n) = n^2/\lg n$$

Si può applicare il Master Theorem????



$n^{\log_b a} = n^2$ vs $n^2/\log n$

Caso 1 ($n^{2-\varepsilon}$)**, osserva:**

n^{ε} ???? $\log n, \forall \varepsilon > 0$

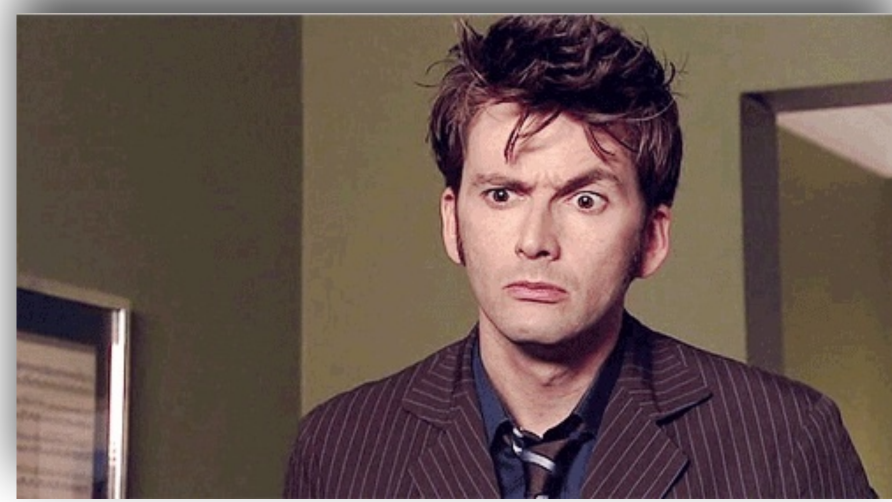
Esercizio 4

Proviamo con il MT….

$$T(n) = 4T(n/2) + n^2/\log n$$

$$a = 4,\ b = 2 \Rightarrow n^{\log_b a} = n^2 \quad \textbf{e} \quad f(n) = n^2/\lg n$$

Si può applicare il Master Theorem????



$n^{\log_b a} = n^2$ vs $n^2/\log n$

Caso 1 ($n^{2-\varepsilon}$), osserva:

$n^{\varepsilon} = \omega(\log n),\ \forall\ \varepsilon > 0$

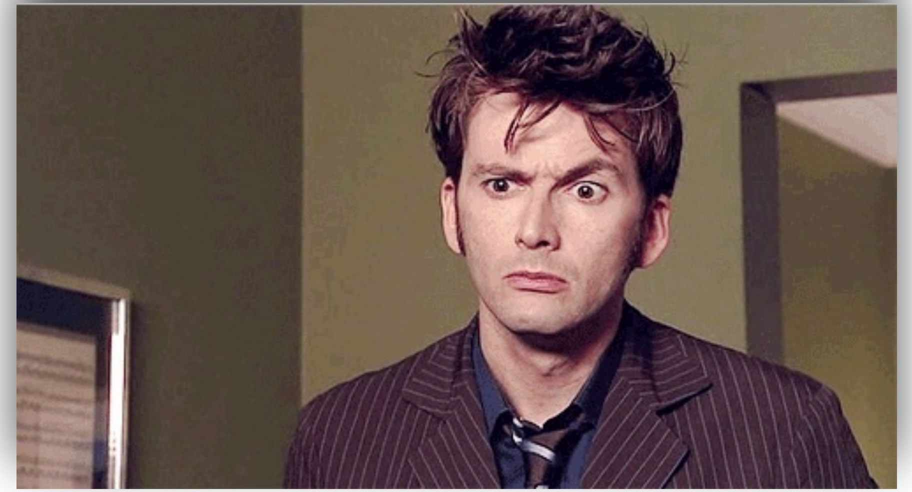
Esercizio 4

Proviamo con il MT....

$$T(n) = 4T(n/2) + n^2/\log n$$

$$a = 4,\ b = 2 \Rightarrow n^{\log_b a} = n^2 \quad \mathbf{e} \quad f(n) = n^2/\lg n$$

Si può applicare il Master Theorem????



$n^{\log_b a} = n^2$ vs $n^2/\log n$

$n^\varepsilon = \omega(\log n),\ \forall\ \varepsilon > 0$

n^2/n^ε ???? $n^2/\log n$

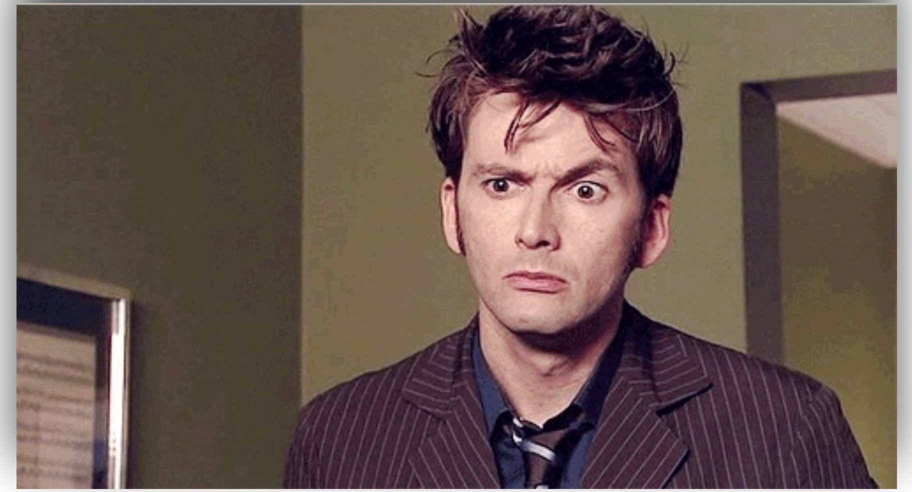
Esercizio 4

Proviamo con il MT….

$$T(n) = 4T(n/2) + n^2/\log n$$

$a = 4,\ b = 2 \Rightarrow n^{\log_b a} = n^2$ **e** $f(n) = n^2/\lg n$

Si può applicare il Master Theorem????



$n^{\log_b a} = n^2$ vs $n^2/\log n$

$n^{\varepsilon} = \omega(\log n),\ \forall\ \varepsilon > 0$

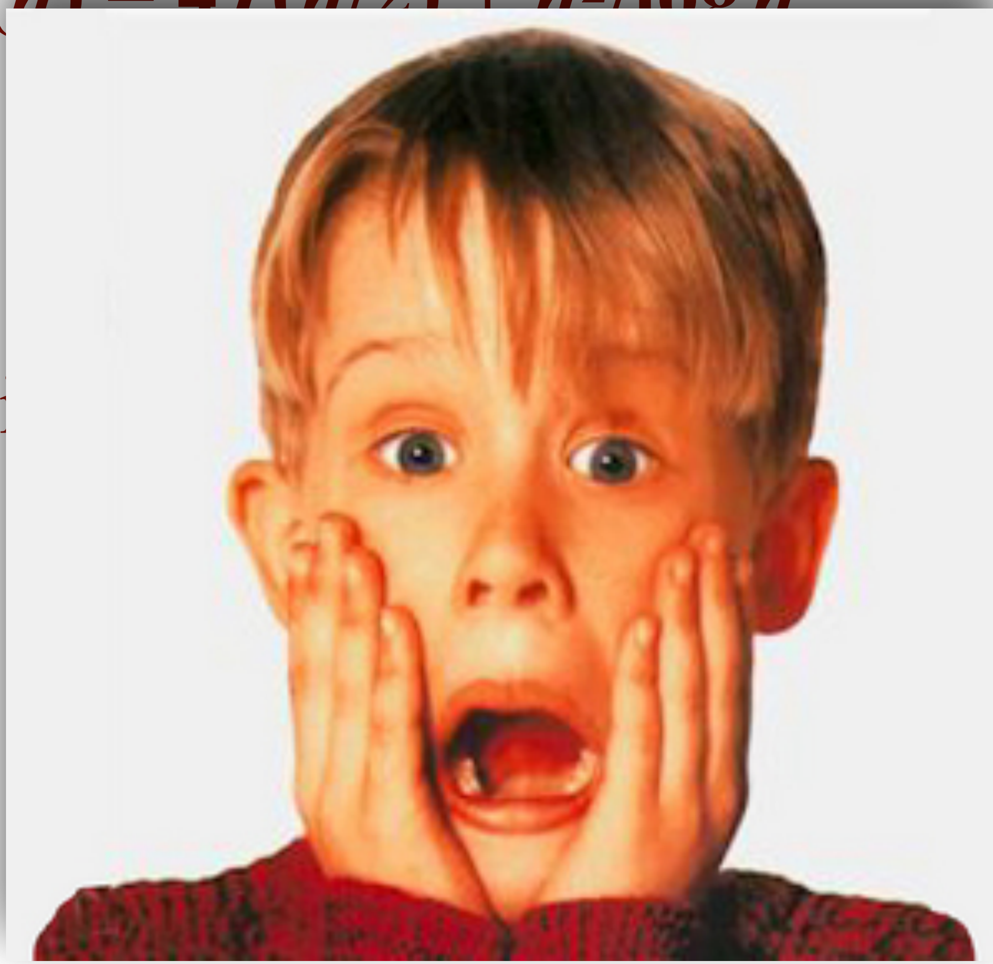
$n^2/n^{\varepsilon} < n^2/\log n$

Caso 1 NO!!!!

Esercizio 4

Proviamo con il MT....

$T(n) = 4T(n/2) + n^2/\log n$



e $f(n) = n^2/\lg n$

S… orem????

$n^{\log_b a} = n^2$ vs $n^2/\log n$

$n^{\varepsilon} = \omega(\log n), \forall\, \varepsilon > 0$

Nessun Caso!!!!

Esercizio 4

E quindi????

$$T(n) = 4T(n/2) + n^2/\log n$$

Esercizio 4

Svolgete....

$$T(n) = 4T(n/2) + n^2/\log n$$

Esercizio 4

Svolgete....

$$T(n) = 4T(n/2) + \frac{n^2}{\log n} =$$

Esercizio 4

Svolgete....

$$T(n) = 4T(n/2) + \frac{n^2}{\log n} = 4(4T(\frac{n}{4}) + \frac{(\frac{n}{2})^2}{\log(n/2)}) + \frac{n^2}{\log n} =$$

Esercizio 4

Svolgete....

$$T(n) = 4T(n/2) + \frac{n^2}{\log n} = 4(4T(\frac{n}{4}) + \frac{(\frac{n}{2})^2}{\log(n/2)}) + \frac{n^2}{\log n} =$$

$$16T(n/4) + \frac{\frac{4n^2}{4}}{\log n/2} + \frac{n^2}{\log n} =$$

Esercizio 4

Svolgete....

$$T(n) = 4T(n/2) + \frac{n^2}{\log n} = 4(4T(\frac{n}{4}) + \frac{(\frac{n}{2})^2}{\log(n/2)}) + \frac{n^2}{\log n} =$$

$$16T(n/4) + \frac{\frac{4n^2}{4}}{\log n/2} + \frac{n^2}{\log n} =$$

$$16T(n/4) + \frac{n^2}{\log n/2} + \frac{n^2}{\log n} = \ldots =$$

Esercizio 4

Svolgete….

$$T(n) = 4T(n/2) + \frac{n^2}{\log n} = 4(4T(\frac{n}{4}) + \frac{(\frac{n}{2})^2}{\log(n/2)}) + \frac{n^2}{\log n} =$$

$$16T(n/4) + \frac{\frac{4n^2}{4}}{\log n/2} + \frac{n^2}{\log n} =$$

$$16T(n/4) + \frac{n^2}{\log n/2} + \frac{n^2}{\log n} = \ldots =$$

$$2^{2k}T(n/2^k) + n^2\Sigma_{i=0}^{k-1}\frac{1}{\log(\frac{n}{2^i})} =$$

Esercizio 4

Svolgete....

$$T(n) = 4T(n/2) + \frac{n^2}{\log n} = 4(4T(\frac{n}{4}) + \frac{(\frac{n}{2})^2}{\log(n/2)}) + \frac{n^2}{\log n} =$$

$$16T(n/4) + \frac{\frac{4n^2}{4}}{\log n/2} + \frac{n^2}{\log n} =$$

$$16T(n/4) + \frac{n^2}{\log n/2} + \frac{n^2}{\log n} = \ldots =$$

$$2^{2k}T(n/2^k) + n^2 \Sigma_{i=0}^{k-1} \frac{1}{\log(\frac{n}{2^i})} = [n/2^k = 1 \leftrightarrow k = \log n]$$

Esercizio 4

Svolgete....

$$T(n) = 4T(n/2) + \frac{n^2}{\log n} = 4(4T(\frac{n}{4}) + \frac{(\frac{n}{2})^2}{\log(n/2)}) + \frac{n^2}{\log n} =$$

$$16T(n/4) + \frac{\frac{4n^2}{4}}{\log n/2} + \frac{n^2}{\log n} =$$

$$16T(n/4) + \frac{n^2}{\log n/2} + \frac{n^2}{\log n} = \ldots =$$

$$2^{2k}T(n/2^k) + n^2\Sigma_{i=0}^{k-1}\frac{1}{\log(\frac{n}{2^i})} = [n/2^k = 1 \leftrightarrow k = \log n]$$

$$n^2 + n^2\Sigma_{i=0}^{k-1}\frac{1}{\log n - \log 2^i} =$$

Esercizio 4

Svolgete....

$$T(n) = 4T(n/2) + \frac{n^2}{\log n} = 4(4T(\frac{n}{4}) + \frac{(\frac{n}{2})^2}{\log(n/2)}) + \frac{n^2}{\log n} =$$

$$16T(n/4) + \frac{\frac{4n^2}{4}}{\log n/2} + \frac{n^2}{\log n} =$$

$$16T(n/4) + \frac{n^2}{\log n/2} + \frac{n^2}{\log n} = \ldots =$$

$$2^{2k}T(n/2^k) + n^2\Sigma_{i=0}^{k-1}\frac{1}{\log(\frac{n}{2^i})} = [n/2^k = 1 \leftrightarrow k = \log n]$$

$$n^2 + n^2\Sigma_{i=0}^{k-1}\frac{1}{\log n - \log 2^i} = n^2 + n^2\Sigma_{i=0}^{k-1}\frac{1}{\log n - i} =$$

Esercizio 4

Svolgete....

$$T(n) = 4T(n/2) + \frac{n^2}{\log n} = 4(4T(\frac{n}{4}) + \frac{(\frac{n}{2})^2}{\log(n/2)}) + \frac{n^2}{\log n} =$$

$$16T(n/4) + \frac{\frac{4n^2}{4}}{\log n/2} + \frac{n^2}{\log n} =$$

$$16T(n/4) + \frac{n^2}{\log n/2} + \frac{n^2}{\log n} = \ldots =$$

$$2^{2k}T(n/2^k) + n^2\Sigma_{i=0}^{k-1}\frac{1}{\log(\frac{n}{2^i})} = [n/2^k = 1 \leftrightarrow k = \log n]$$

$$n^2 + n^2\Sigma_{i=0}^{k-1}\frac{1}{\log n - \log 2^i} = n^2 + n^2\Sigma_{i=0}^{k-1}\frac{1}{\log n - i} =$$

$$n^2 + n^2\Sigma_{i=0}^{k-1}\frac{1}{k-i} =$$

Esercizio 4

Svolgete....

$$T(n) = 4T(n/2) + \frac{n^2}{\log n} = 4(4T(\frac{n}{4}) + \frac{(\frac{n}{2})^2}{\log(n/2)}) + \frac{n^2}{\log n} =$$

$$16T(n/4) + \frac{\frac{4n^2}{4}}{\log n/2} + \frac{n^2}{\log n} =$$

$$16T(n/4) + \frac{n^2}{\log n/2} + \frac{n^2}{\log n} = \ldots =$$

$$2^{2k}T(n/2^k) + n^2\Sigma_{i=0}^{k-1}\frac{1}{\log(\frac{n}{2^i})} = [n/2^k = 1 \leftrightarrow k = \log n]$$

$$n^2 + n^2\Sigma_{i=0}^{k-1}\frac{1}{\log n - \log 2^i} = n^2 + n^2\Sigma_{i=0}^{k-1}\frac{1}{\log n - i} =$$

$$n^2 + n^2\Sigma_{i=0}^{k-1}\frac{1}{k-i} = [\Sigma_{i=0}^{k-1}\frac{1}{k-i} \simeq \log k] =$$

Esercizio 4

Svolgete....

$$T(n) = 4T(n/2) + \frac{n^2}{\log n} = 4(4T(\frac{n}{4}) + \frac{(\frac{n}{2})^2}{\log(n/2)}) + \frac{n^2}{\log n} =$$

$$16T(n/4) + \frac{\frac{4n^2}{4}}{\log n/2} + \frac{n^2}{\log n} =$$

$$16T(n/4) + \frac{n^2}{\log n/2} + \frac{n^2}{\log n} = \ldots =$$

$$2^{2k}T(n/2^k) + n^2\Sigma_{i=0}^{k-1}\frac{1}{\log(\frac{n}{2^i})} = [n/2^k = 1 \leftrightarrow k = \log n]$$

$$n^2 + n^2\Sigma_{i=0}^{k-1}\frac{1}{\log n - \log 2^i} = n^2 + n^2\Sigma_{i=0}^{k-1}\frac{1}{\log n - i} =$$

$$n^2 + n^2\Sigma_{i=0}^{k-1}\frac{1}{k-i} = [\Sigma_{i=0}^{k-1}\frac{1}{k-i} \simeq \log k] = O(n^2 \log\log n)$$

Esercizio 5

Descrivere un algoritmo di costo **Θ(nlogn)** che, dato un insieme S di n numeri interi e un altro intero x, determini se esistono due elementi in S la cui somma è esattamente x

Esempio:

$S = \{4, 7, -1, 9, 4, 10, 22, -34\}$

$x = 8$

Esercizio 5

Descrivere un algoritmo di costo **Θ(nlogn)** che, dato un insieme S di n numeri interi e un altro intero x, determini se esistono due elementi in S la cui somma è esattamente x

Esempio:

$S = \{4, 7, -1, 9, 4, 10, 22, -34\}$

$x = 8$

Risultato: True [ad es., (4, 4) o (-1, 9)]

$S = \{4, 7, -1, 9, 4, 10, 22, -34\}$ $\qquad x = 8$

1. Ordina gli elementi in S

$S = \{-34, -1, 4, 4, 7, 9, 10, 22\}$ $x = 8$

1. Ordina gli elementi in S

$S = \{-34, -1, 4, 4, 7, 9, 10, 22\}$ $x = 8$

1. Ordina gli elementi in S
2. Per ogni $y \in S$, controlla che $x - y \in S$

$S = \{\mathbf{-34}, -1, 4, 4, 7, 9, 10, 22\}$ $x = 8$

$8 + 34 = 42$: NO

1. Ordina gli elementi in S
2. Per ogni y ∈ S, controlla che x - y ∈ S

$S = \{-34, \mathbf{-1}, 4, 4, 7, 9, 10, 22\}$ $x = 8$

$8 + 1 = 9$: **SI**

1. Ordina gli elementi in S
2. Per ogni $y \in S$, controlla che $x - y \in S$

$S = \{-34, \mathbf{-1}, 4, 4, 7, 9, 10, 22\}$ $x = 8$

$8 + 1 = 9$: **SI**

Tempo passo 1: ????
Tempo passo 2: ????

1. Ordina gli elementi in S
2. Per ogni $y \in S$, controlla che $x - y \in S$

$S = \{-34, \mathbf{-1}, 4, 4, 7, 9, 10, 22\}$ $x = 8$

$8 + 1 = 9$: **SI**

Tempo passo 1: O(n log n)
Tempo passo 2: ????

1. Ordina gli elementi in S
2. Per ogni $y \in S$, controlla che $x - y \in S$

S = {-34, **-1**, 4, 4, 7, 9, 10, 22} x = 8

8 + 1 = 9: **SI**

Tempo passo 1: O(n log n)
Tempo passo 2: O(n log n)

1. Ordina gli elementi in S
2. Per ogni y ∈ S, controlla che x - y ∈ S

$S = \{-34, \mathbf{-1}, 4, 4, 7, 9, 10, 22\}$ $x = 8$

8 + 1 = 9: **SI**

Ricerca Binaria

Tempo passo 1: O(n log n)
Tempo passo 2: O(n log n)

1. Ordina gli elementi in S
2. Per ogni $y \in S$, controlla che $x - y \in S$

$S = \{-34, -1, 4, 4, 7, 9, 10, 22\}$ $\quad x = 8$

Altra soluzione (leggermente migliore) ????

1. Ordina gli elementi in S

$S = \{-34, -1, 4, 4, 7, 9, 10, 22\}$ $x = 8$

i j

Altra soluzione (leggermente migliore) ????

1. Ordina gli elementi in S

$S = \{-34, -1, 4, 4, 7, 9, 10, 22\}$ $x = 8$

i j

Altra soluzione (leggermente migliore) ????

1. Ordina gli elementi in S
2. Se S[i] + S[j] == x: True

$S = \{-34, -1, 4, 4, 7, 9, 10, 22\}$ $\qquad x = 8$

i $\qquad\qquad$ j

Altra soluzione (leggermente migliore) ????

1. Ordina gli elementi in S
2. Se $S[i] + S[j] == x$: True
3. Se $S[i] + S[j] < x$: i++

$$S = \{-34, -1, 4, 4, 7, 9, 10, 22\} \qquad x = 8$$

i j

$-34 + 8 = -26$

Altra soluzione (leggermente migliore) ????

1. Ordina gli elementi in S
2. Se S[i] + S[j] == x: True
3. Se S[i] + S[j] < x: i++

$S = \{-34, -1, 4, 4, 7, 9, 10, 22\}$ $\qquad x = 8$

i $\qquad$ j

$-1 + 22 = 21$

Altra soluzione (leggermente migliore) ????

1. Ordina gli elementi in S
2. Se S[i] + S[j] == x: True
3. Se S[i] + S[j] < x: i++
4. Altrimenti: j- -

$S = \{-34, -1, 4, 4, 7, 9, 10, 22\}$ $\quad$ $x = 8$

i $\quad$ j

$-1 + 22 = 21$

Altra soluzione (leggermente migliore) ????

1. Ordina gli elementi in S
2. Se S[i] + S[j] == x: True
3. Se S[i] + S[j] < x: i++
4. Altrimenti: j- -

$S = \{-34, -1, 4, 4, 7, 9, 10, 22\}$ $\quad x = 8$

$-1 + 10 = 9$

i j

Altra soluzione (leggermente migliore) ????

1. Ordina gli elementi in S
2. Se S[i] + S[j] == x: True
3. Se S[i] + S[j] < x: i++
4. Altrimenti: j- -

$S = \{-34, -1, 4, 4, 7, 9, 10, 22\}$ $\qquad$ $x = 8$

$-1 + 10 = 9$

i $\qquad\qquad$ j

Altra soluzione (leggermente migliore) ????

1. Ordina gli elementi in S
2. Se S[i] + S[j] == x: True
3. Se S[i] + S[j] < x: i++
4. Altrimenti: j- -

$S = \{-34, -1, 4, 4, 7, 9, 10, 22\}$ $x = 8$

i j

-1 + 9 = 8

Altra soluzione (leggermente migliore) ????

1. Ordina gli elementi in S
2. Se S[i] + S[j] == x: True
3. Se S[i] + S[j] < x: i++
4. Altrimenti: j- -

$S = \{-34, -1, 4, 4, 7, 9, 10, 22\}$ $\quad x = 8$

i j

-1 + 9 = 8

Altra soluzione (leggermente migliore) ????

1. Ordina gli elementi in S
2. Se S[i] + S[j] == x: True
3. Se S[i] + S[j] < x: i++
4. Altrimenti: j- -

Tempo passo 1: O(n log n)
Tempo passi 2-4: ????

S = {-34, -1, 4, 4, 7, 9, 10, 22} x = 8

i   j

-1 + 9 = 8

Altra soluzione (leggermente migliore) ????

1. Ordina gli elementi in S
2. Se S[i] + S[j] == x: True
3. Se S[i] + S[j] < x: i++
4. Altrimenti: j- -

Tempo passo 1: O(n log n)
Tempo passi 2-4: O(n)

Esercizio 6

Siano:

- $T(n) = 7T(n/2) + n^2$ una funzione di costo di un algoritmo A, **e**
- $T'(n) = aT'(n/4) + n^2$ una funzione di costo di un algoritmo A'

Qual è il massimo valore intero di a che rende A' asintoticamente più veloce di A?

Siano:

- $T(n) = 7T(n/2) + n^2$ una funzione di costo di un algoritmo A, **e**
- $T'(n) = aT'(n/4) + n^2$ una funzione di costo di un algoritmo A'

Qual è il massimo valore intero di a che rende A' asintoticamente più veloce di A?

Siano:

- $T(n) = 7T(n/2) + n^2$ una funzione di costo di un algoritmo A, **e**
- $T'(n) = aT'(n/4) + n^2$ una funzione di costo di un algoritmo A'

Qual è il massimo valore intero di a che rende A' asintoticamente più veloce di A?

$$T(n) = ????$$

Siano:

- $T(n) = 7T(n/2) + n^2$ una funzione di costo di un algoritmo A, **e**
- $T'(n) = aT'(n/4) + n^2$ una funzione di costo di un algoritmo A'

Qual è il massimo valore intero di a che rende A' asintoticamente più veloce di A?

$$T(n) = ????$$

$$\log_2 7 \cong 2.8 \rightarrow$$

Siano:

- T (n) = 7T (n/2) + n^2 una funzione di costo di un algoritmo A, **e**
- T '(n) = aT '(n/4) + n^2 una funzione di costo di un algoritmo A'

Qual è il massimo valore intero di a che rende A' asintoticamente più veloce di A?

$$T(n) = ????$$

$$\log_2 7 \tilde{=} 2.8 \rightarrow n^2 = O(n^{\log_2 7 - \epsilon})$$

Siano:

- T (n) = 7T (n/2) + n^2 una funzione di costo di un algoritmo A, **e**
- T '(n) = aT '(n/4) + n^2 una funzione di costo di un algoritmo A'

Qual è il massimo valore intero di a che rende A' asintoticamente più veloce di A?

$$T(n) = O(n^{\log_2 7})$$

$$\log_2 7 \cong 2.8 \rightarrow n^2 = O(n^{\log_2 7 - \epsilon})$$

Siano:

- T (n) = 7T (n/2) + n^2 una funzione di costo di un algoritmo A, **e**
- T '(n) = aT '(n/4) + n^2 una funzione di costo di un algoritmo A'

Qual è il massimo valore intero di a che rende A' asintoticamente più veloce di A?

$$T(n) = O(n^{\log_2 7})$$

$$\log_2 7 \simeq 2.8 \rightarrow n^2 = O(n^{\log_2 7 - \epsilon})$$

$$\log_2 7 = \frac{\log_4 7}{\log_4 2} = \frac{\log_4 7}{1/2} = 2 \log_4 7 = \log_4 49$$

Siano:

- T (n) = 7T (n/2) + n^2 una funzione di costo di un algoritmo A, **e**
- T '(n) = aT '(n/4) + n^2 una funzione di costo di un algoritmo A'

Qual è il massimo valore intero di a che rende A' asintoticamente più veloce di A?

$$T(n) = O(n^{\log_2 7})$$

$$\log_2 7 \simeq 2.8 \to n^2 = O(n^{\log_2 7 - \epsilon})$$

$$\log_2 7 = \log_4 49$$

Siano:

- $T(n) = 7T(n/2) + n^2$ una funzione di costo di un algoritmo A, **e**
- $T'(n) = aT'(n/4) + n^2$ una funzione di costo di un algoritmo A'

Qual è il massimo valore intero di a che rende A' asintoticamente più veloce di A?

$$T(n) = O(n^{\log_2 7})$$

$$\log_2 7 = \log_4 49$$

$$T'(n) = ????$$

$$n^{\log_4 a} \;????\; n^2$$

Siano:

- T (n) = 7T (n/2) + n^2 una funzione di costo di un algoritmo A, **e**
- T $'$(n) = aT $'$(n/4) + n^2 una funzione di costo di un algoritmo A$'$

Qual è il massimo valore intero di a che rende A$'$ asintoticamente più veloce di A?

$$T(n) = O(n^{\log_2 7})$$

$$\log_2 7 = \log_4 49$$

$$T'(n) = ????$$

$$n^{\log_4 a} \; ???? \; n^2$$

$\log_4 16 = 2$

Siano:

- T (n) = 7T (n/2) + n^2 una funzione di costo di un algoritmo A, **e**
- T '(n) = aT '(n/4) + n^2 una funzione di costo di un algoritmo A'

Qual è il massimo valore intero di a che rende A' asintoticamente più veloce di A?

$$T(n) = O(n^{\log_2 7})$$

$$\log_2 7 = \log_4 49$$

$\log_4 16 = 2$

$$T'(n) = ????$$

$$n^{\log_4 a} \; ???? \; n^2$$

Se a < 16: T'(n) = ????

Siano:

- T (n) = 7T (n/2) + n^2 una funzione di costo di un algoritmo A, **e**
- T '(n) = aT '(n/4) + n^2 una funzione di costo di un algoritmo A'

Qual è il massimo valore intero di a che rende A' asintoticamente più veloce di A?

$$T(n) = O(n^{\log_2 7})$$

$$\log_2 7 = \log_4 49$$

$$T'(n) = ????$$

$$n^{\log_4 a} \ ???? \ n^2$$

$\log_4 16 = 2$

$$n^{\log_4 a} < n^2$$

Se a < 16: T'(n) = ????

Siano:

- T (n) = 7T (n/2) + n^2 una funzione di costo di un algoritmo A, **e**
- T '(n) = aT '(n/4) + n^2 una funzione di costo di un algoritmo A'

Qual è il massimo valore intero di a che rende A' asintoticamente più veloce di A?

$$T(n) = O(n^{\log_2 7})$$

$$\log_2 7 = \log_4 49$$

$\log_4 16 = 2$

$$T'(n) = ????$$

$$n^{\log_4 a} \; ???? \; n^2$$

$$n^{\log_4 a} < n^2$$

Se a < 16: T'(n) = θ(n^2)

Siano:

- T (n) = 7T (n/2) + n^2 una funzione di costo di un algoritmo A, **e**
- T '(n) = aT '(n/4) + n^2 una funzione di costo di un algoritmo A'

Qual è il massimo valore intero di a che rende A' asintoticamente più veloce di A?

$$T(n) = O(n^{\log_2 7})$$

$$\log_2 7 = \log_4 49$$

$$T'(n) = ????$$

$$n^{\log_4 a} \; ???? \; n^2$$

$\log_4 16 = 2$

$$n^{\log_4 a} < n^2$$

Se a < 16: T'(n) = $\theta(n^2)$

$$n^{\log_4 a} = n^2$$

Se a = 16: T'(n) = ????

Siano:

- $T(n) = 7T(n/2) + n^2$ una funzione di costo di un algoritmo A, **e**
- $T'(n) = aT'(n/4) + n^2$ una funzione di costo di un algoritmo A'

Qual è il massimo valore intero di a che rende A' asintoticamente più veloce di A?

$$T(n) = O(n^{\log_2 7})$$

$$\log_2 7 = \log_4 49$$

$$T'(n) = ????$$

$$n^{\log_4 a} \ ???? \ n^2$$

$\log_4 16 = 2$

$$T'(n) = O(T(n)) \left| \begin{array}{l} n^{\log_4 a} < n^2 \\ n^{\log_4 a} = n^2 \end{array} \right.$$

Se $a < 16$: $T'(n) = \theta(n^2)$

Se $a = 16$: $T'(n) = n^2 \log n$

Siano:

- T (n) = 7T (n/2) + n^2 una funzione di costo di un algoritmo A, **e**
- T '(n) = aT '(n/4) + n^2 una funzione di costo di un algoritmo A'

Qual è il massimo valore intero di a che rende A' asintoticamente più veloce di A?

$$T(n) = O(n^{\log_2 7})$$

$$\log_2 7 = \log_4 49$$

$$T'(n) = ????$$

$$n^{\log_4 a} \ ???? \ n^2$$

$\log_4 16 = 2$

$$T'(n) = O(T(n)) \quad \begin{cases} n^{\log_4 a} < n^2 \\ n^{\log_4 a} = n^2 \end{cases}$$

Se a < 16: T'(n) = $\theta(n^2)$

Se a = 16: T'(n) = $n^2 \log n$

Se a > 16, T'(n) = ????

Siano:

- T (n) = 7T (n/2) + n^2 una funzione di costo di un algoritmo A, **e**
- T '(n) = aT '(n/4) + n^2 una funzione di costo di un algoritmo A'

Qual è il massimo valore intero di a che rende A' asintoticamente più veloce di A?

$$T(n) = O(n^{\log_2 7})$$

$$\log_2 7 = \log_4 49$$

$\log_4 16 = 2$

$$T'(n) = ????$$

$$n^{\log_4 a} \;????\; n^2$$

$$T'(n) = O(T(n)) \left| \begin{array}{l} n^{\log_4 a} < n^2 \\ n^{\log_4 a} = n^2 \end{array} \right.$$

Se a < 16: T'(n) = $\theta(n^2)$

Se a = 16: T'(n) = $n^2 \log n$

Se a > 16, T'(n) = $O(n^{\log_4 a})$

Siano:

- T (n) = 7T (n/2) + n^2 una funzione di costo di un algoritmo A, **e**
- T $'$(n) = aT $'$(n/4) + n^2 una funzione di costo di un algoritmo A$'$

Qual è il massimo valore intero di a che rende A$'$ asintoticamente più veloce di A?

$$T(n) = O(n^{\log_2 7})$$

$$\log_2 7 = \log_4 49$$

$$T'(n) = ????$$

$$n^{\log_4 a} \; ???? \; n^2$$

$\log_4 16 = 2$

$$T'(n) = O(T(n)) \quad \left| \begin{array}{l} n^{\log_4 a} < n^2 \\ n^{\log_4 a} = n^2 \end{array} \right.$$

Se a < 16: T'(n) = θ(n^2)

Se a = 16: T'(n) = n^{2}log n

Se a > 16, T'(n) = O(n$^{\log_4 a}$)

Quando $T'(n) = O(T(n))$????

Siano:

- T (n) = 7T (n/2) + n^2 una funzione di costo di un algoritmo A, **e**
- T '(n) = aT '(n/4) + n^2 una funzione di costo di un algoritmo A'

Qual è il massimo valore intero di a che rende A' asintoticamente più veloce di A?

$$T(n) = O(n^{\log_2 7})$$

$$\log_2 7 = \log_4 49$$

$\log_4 16 = 2$

$$T'(n) = ????$$

$$n^{\log_4 a} \; ???? \; n^2$$

$$T'(n) = O(T(n)) \quad \begin{array}{|l} n^{\log_4 a} < n^2 \\ n^{\log_4 a} = n^2 \end{array}$$

Se a < 16: T'(n) = $\theta(n^2)$

Se a = 16: T'(n) = $n^2 \log n$

Se a > 16, T'(n) = $O(n^{\log_4 a})$

Quando $T'(n) = O(T(n))$????

a < 49