⟨Teorema Maestro⟩

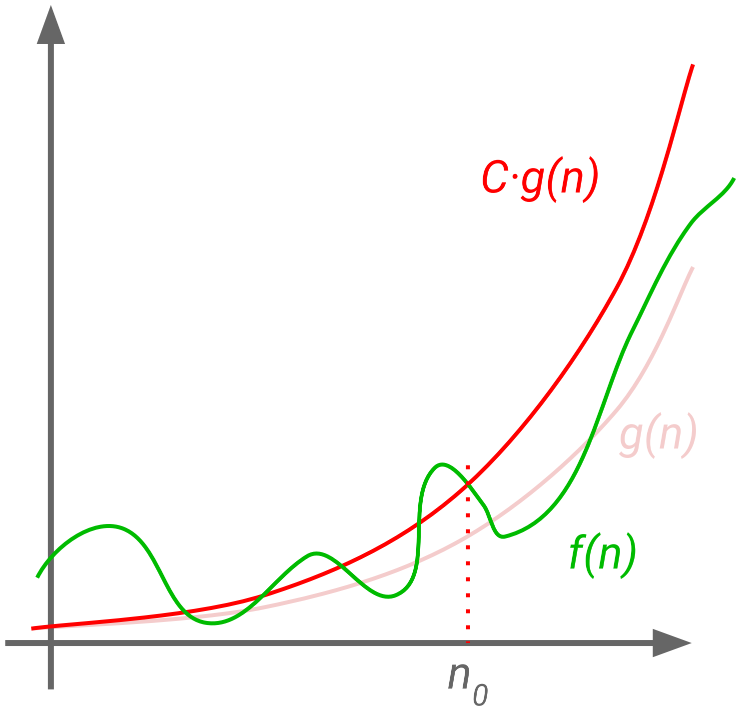
CC3001 2021-1 - Auxiliar 3

Notación O

O grande

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

$$(\exists C)(\exists\, n_0)(\forall n > n_0)|f(n)| \leq C \cdot g(n)$$

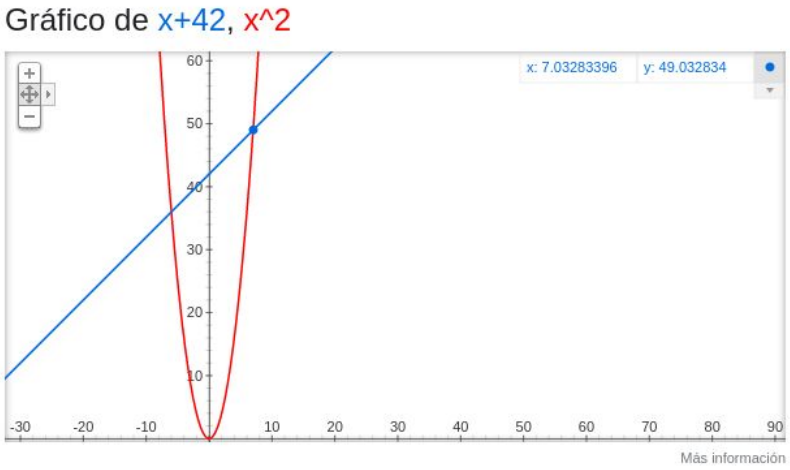


Una función *f* es “O grande de g” si,

a partir de un punto,

está acotada por arriba por *g* (o por *g* “estirado” por una constante cualquiera)

O grande - Ejemplo



$f(n) = n+42$ es $O(n^2)$, porque a partir de $n_0=7$:

$$|n + 42| \leq 1 \cdot n^2$$

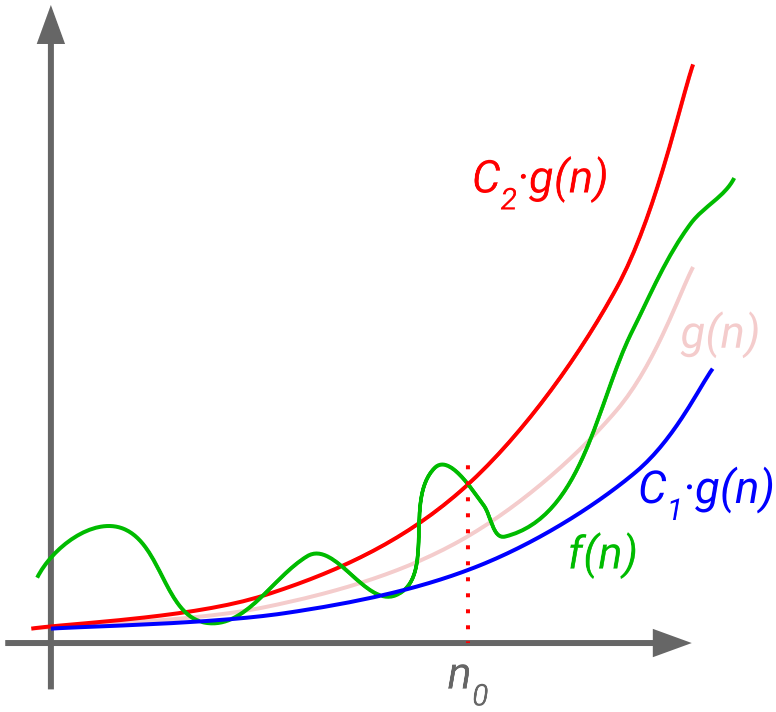
Theta grande

$$f(n) \in \Theta(g(n)) \iff$$
$$(\exists C_1, C_2)(\exists\, n_0)(\forall n > n_0)$$
$$C_1 \cdot g(n) \ \leq |f(n)| \leq \ C_2 \cdot g(n)$$

Una función *f* es “Theta grande de g” si,

a partir de un punto,

está acotada tanto por arriba como por abajo por versiones “estiradas” de g.



Theta grande - Ejemplo

Gráfico de x*log2(x)+5, 2*x*log2(x), x*log2(x)

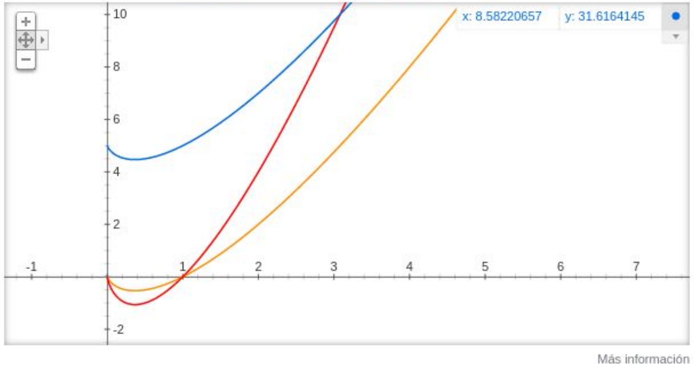
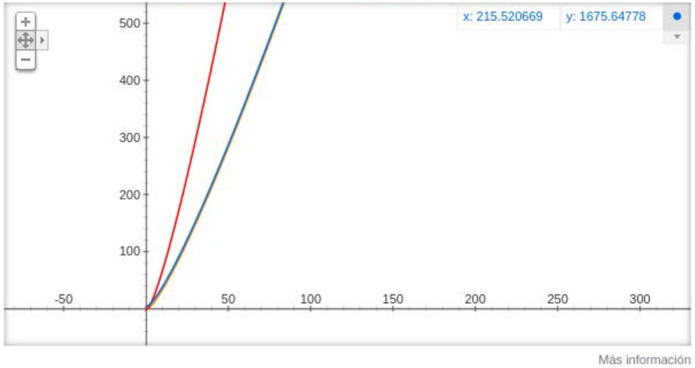


Gráfico de x*log2(x)+5, 2*x*log2(x), x*log2(x)



$f(n) = n \cdot log_2 n+5$ es $\Theta(n \cdot log_2 n)$, porque

a partir de $n_0=4$

$$1 \cdot n \cdot \log_2(n) \leq |n \cdot \log_2(n) + 5| \leq 2 \cdot n \cdot \log_2(n)$$

Otros en:

https://en.wikipedia.org/wiki/Big_O_notation#Family_of_Bachmann%E2%80%93Landau_notations

El teorema

Teorema MaestrO - Dividir (y unir) para Reinar

$$f(n) = a \cdot f\left(\frac{n}{b}\right) + O(n^c)$$

El problema f(n)

...lo divido en ***a*** subproblemas de tamaño ***n/b***

...y luego tengo que hacer un trabajo ***O(n^c)*** para unir los subresultados y entregar un resultado

(O tengo que hacer un trabajo previo para preparar los subproblemas)

Teorema MaestrO

Si $f(n) = a \cdot f(\frac{n}{b}) + O(n^c)$, $a \geq 1, b > 1$, entonces:

$$f(n) \in \begin{cases} \Theta(n^c) & \text{si } a < b^c \\ \Theta(n^c \cdot \log(n)) & \text{si } a = b^c \\ \Theta(n^{\log_b(a)}) & \text{si } a > b^c \end{cases}$$

Caso $a<b^c$

Teorema Maestro - $a < b^c$

$$f(n) = a \cdot f\left(\frac{n}{b}\right) + O(n^c)$$

Si el término ***O(n^c^)*** es una constante:

- ***$O(n^c)$*** es ***$k \cdot n^0$*** -> ***$c = 0$***.
- ***$a < b^c$ -> $a < b^0$ -> $a < 1$***
- Pero ***$a \geq 1$*** era una hipótesis del ™
- ***$O(n^c)$ no puede ser constante***

El trabajo ***$O(n^c)$*** es significativo, termina dominando el crecimiento de la función:

$a < b^c$ -> $f(n)$ es $\Theta(n^c)$

Caso $a = b^c$

Teorema Maestro - $a = b^c$

$$f(n) = a \cdot f\left(\frac{n}{b}\right) + O(n^c)$$

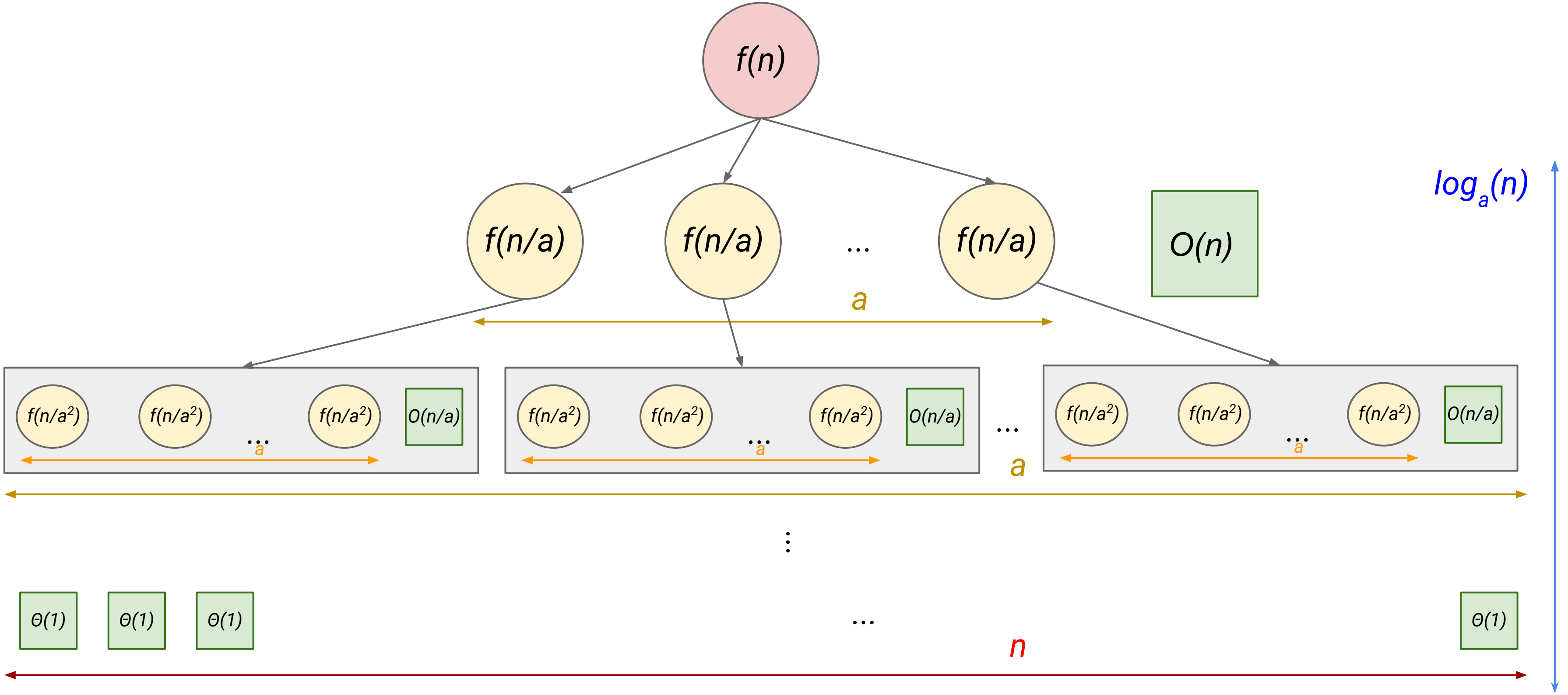
Si ***c***=1

$$f(n) = a \cdot f\left(\frac{n}{b}\right) + O(n)$$

-> ***a*** = ***b***

Teorema Maestro - $a = b$

$$f(n) = a \cdot f\left(\frac{n}{a}\right) + O(n)$$



Teorema Maestro - $a = b$

$$f(n) = a \cdot f(\tfrac{n}{a}) + \boxed{O(n)}$$

$log_a(n)$

f(n)

O(n)

a·O(n/a)* = *O(n)

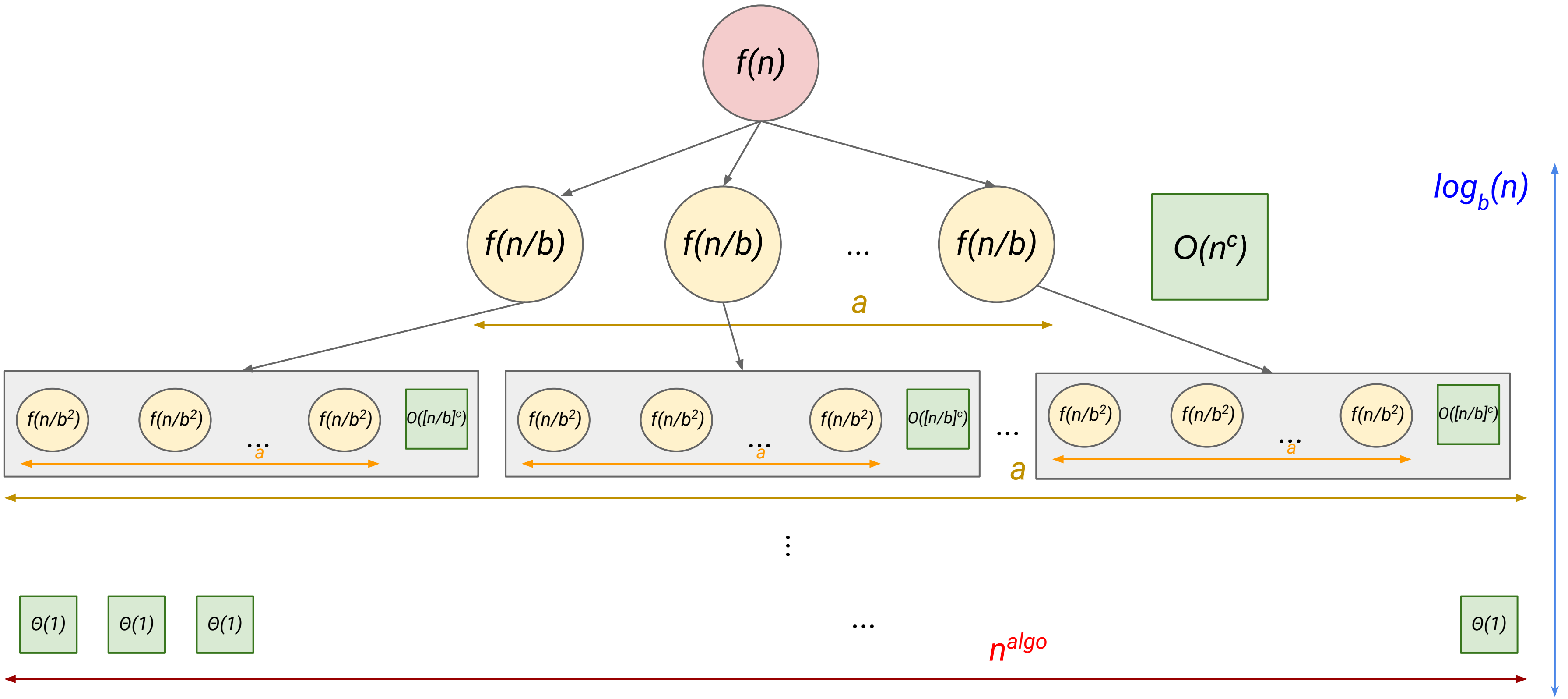
⋮

n·O(1)* = *O(n)

Si ***c***=1, ***a = b*** -> ***f(n)*** es ***O(n·log n)***

Teorema Maestro - $a = b^c$

$$f(n) = a \cdot f(\tfrac{n}{b}) + O(n^c)$$



Teorema Maestro - $a = b^c$

$$f(n) = a \cdot f(\tfrac{n}{a}) + \boxed{O(n)}$$

$f(n)$

$O(n^c)$

$a \cdot O([n/b]^c) = [a/b^c] \cdot O(n^c)$

$a^2 \cdot O([n/b^2]^c) = [a/b^c]^2 \cdot O(n^c)$

$\vdots$

$a^{algo} \cdot O([n/b^p]^c) = [a/b^c]^{algo} \cdot O(n^c)$

$log_b(n)$

Como $\boldsymbol{a=b^c}$, entonces $a/b^c = 1$

-> $\boldsymbol{f(n)}$ *es* $\boldsymbol{O(n^c \cdot log(n))}$

Caso $a > b^c$

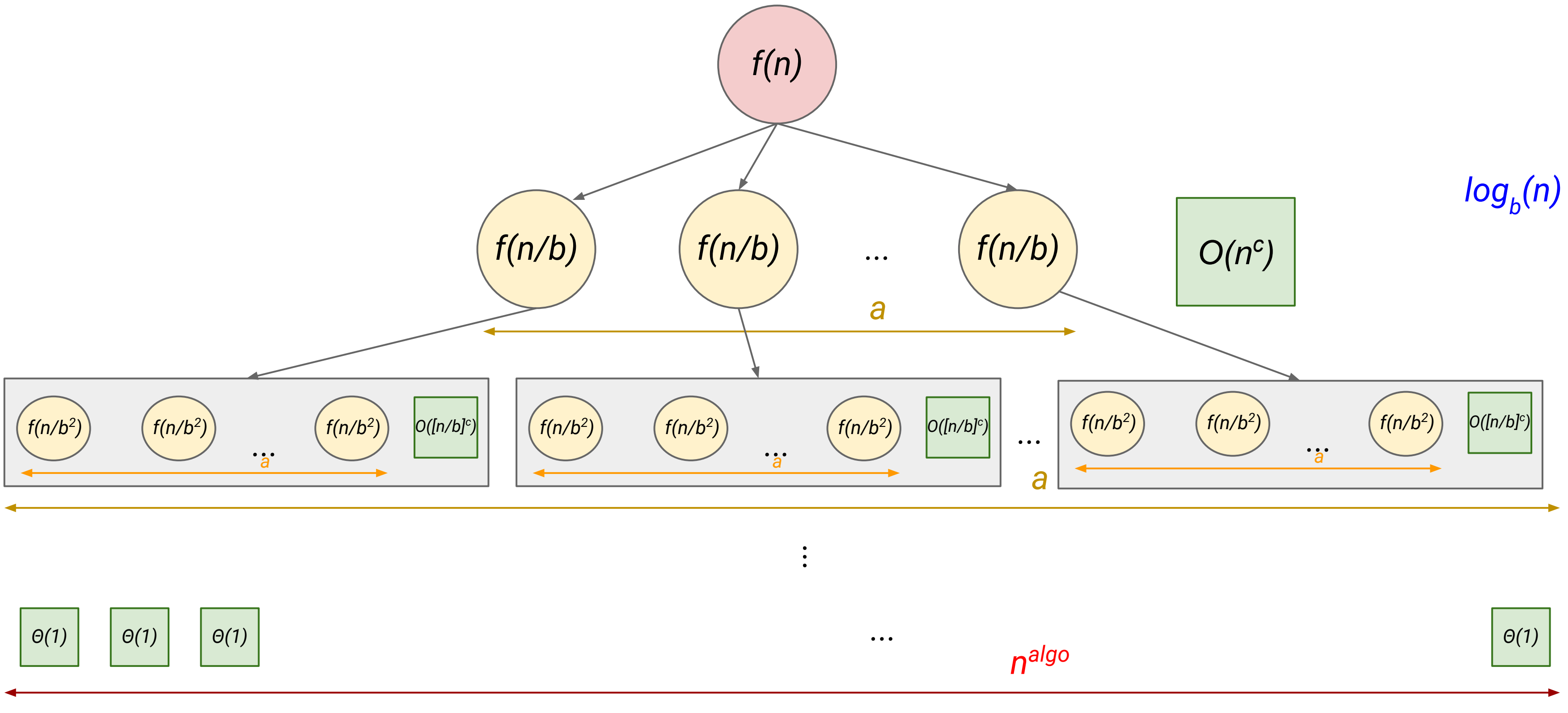
Teorema Maestro - $a > b^c$

$$f(n) = \boxed{a} \cdot f\left(\frac{n}{\boxed{b}}\right) + O(n^{\boxed{c}})$$

Lo que domina el orden es la cantidad de subproblemas, más que el trabajo que hay que hacer en cada nivel.

Teorema Maestro - $a > b^c$

$$f(n) = a \cdot f(\tfrac{n}{b}) + O(n^c)$$



Teorema Maestro - $a > b^c$

$$f(n) = a \cdot f(\tfrac{n}{b}) + O(n^c)$$

Cantidad de hojas:
a·a·a·a·…·a (altura del árbol veces):

#hojas = $a^{log\_b(n)}$

Cambio de base del logaritmo:

$$\log_b(n) = \frac{\log(n)}{\log(b)}$$

entonces el número de hojas es:

$$a^{\log_b(n)} = a^{\frac{\log_a(n)}{\log_a(b)}} = \left(a^{\log_a(n)}\right)^{1/\log_a(b)} = \left(a^{\log_a(n)}\right)^{\log_b(a)}$$

$$\boxed{= n^{\log_b(a)}}$$

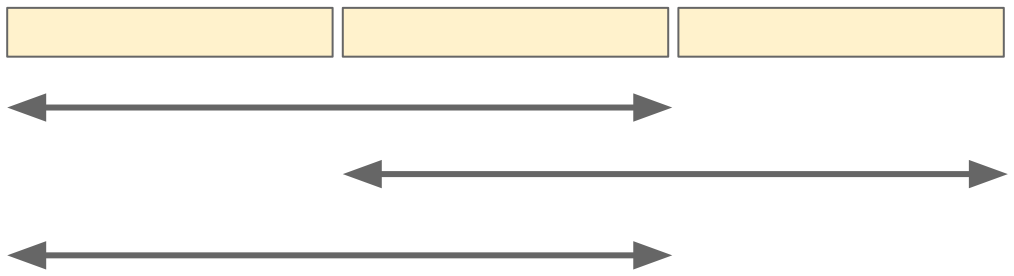
Orden de ***f(n)*** para ***a > b^c***

</Teorema Maestro>

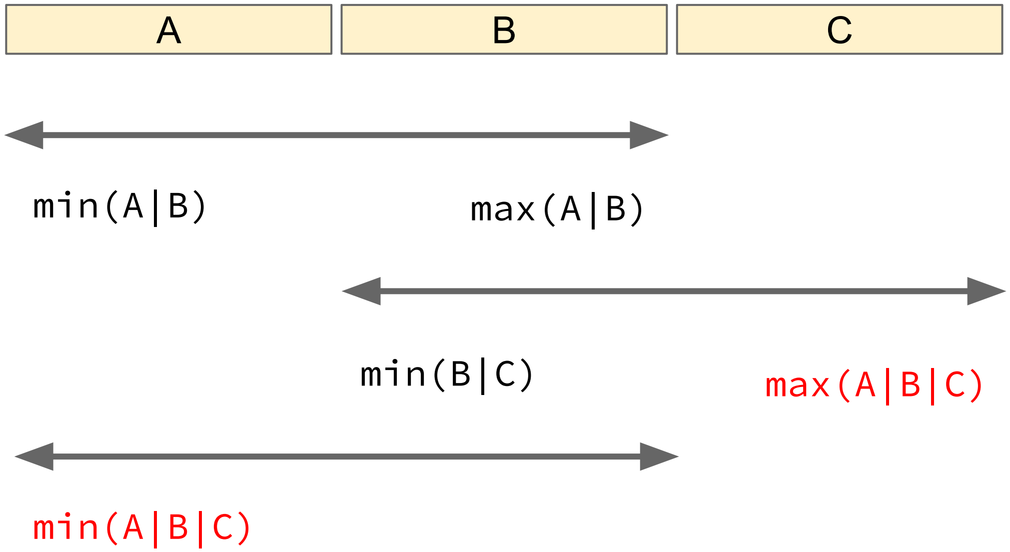
StoogeSort

StoogeSort

- Si el valor en el índice 0 es mayor que el valor en el último índice, entonces intercambia ambos valores.
- Ordena los primeros 2/3 del arreglo con StoogeSort
- Ordena los últimos 2/3 del arreglo con StoogeSort
- Ordena los primeros 2/3 del arreglo con StoogeSort



StoogeSort



StoogeSort

- Si el valor en el índice 0 es mayor que el valor en el último índice, entonces intercambia ambos valores.
- Ordena los primeros 2/3 del arreglo con StoogeSort
- Ordena los últimos 2/3 del arreglo con StoogeSort
- Ordena los primeros 2/3 del arreglo con StoogeSort

$$f(n) = a \cdot f\left(\frac{n}{b}\right) + O(n^c)$$

StoogeSort

- Si el valor en el índice 0 es mayor que el valor en el último índice, entonces intercambia ambos valores.
- Ordena los primeros 2/3 del arreglo con StoogeSort
- Ordena los últimos 2/3 del arreglo con StoogeSort
- Ordena los primeros 2/3 del arreglo con StoogeSort

$$f(n) = a \cdot f\left(\frac{n}{b}\right) + O(n^c)$$