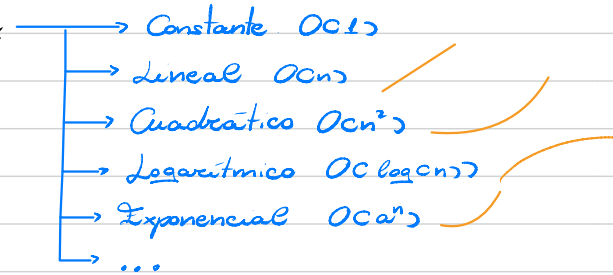


# Tema 1: Eficiencia

## Órdenes de eficiencia



## Análisis de algoritmos (Teórico)

- Sentencias simples  $x += 1 \Rightarrow O(1)$
- Sentencias condicionales  $\text{if } C \text{ Ev. Condición } h \Rightarrow \max\{O(\text{Ev. Condición}), O(B1), O(B2)\}$   
 Bloque 1  
 $\downarrow$  else  $h$   
 Bloque 2  
 $\downarrow$   $g(n)$
- Sentencias repetitivas  $\text{Mientras } C \text{ Condición, hacer } \downarrow h(n) \text{ veces} \Rightarrow O(g(n) + h(n)C(g(n) + f(n))$   
 Bloque  $\rightarrow f(n)$
- Secuencia de sentencias  $\text{Sentencia 1}$   
 $\text{Sentencia 2} \Rightarrow \max\{O(S_1), O(S_2), \dots, O(S_k)\}$   
 $\text{Sentencia } k$

## Eficiencia de una función recursiva

Se calcula el tiempo de exec  $T(n)$  como ecuación en recurrencias

Ejemplo:  $\text{factorial}(n)$

$\text{if } n \leq 1 \text{ return } 1;$   
 $\text{else return } n * \text{factorial}(n-1)$

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

Resolución:  $\rightarrow$  Desarrollo en serie "a lo bruto"

$\rightarrow$  Ecuación característica

$\rightarrow$  Es lineal  $T(n) = a_1 T(n-1) + a_2 T(n-2) + \dots + a_k T(n-k)$

$$T(n) = t_n = x^n$$

\* No homogéneas  $\Rightarrow p(x) = b^n q(x)$   $q(x) = d$   
 $\Rightarrow c(x-b)^{d+1}$

$\rightarrow$  No lineales  $\Rightarrow$  Cambio de variable  $\Rightarrow n = 2^m, 3^m, \log(m)$   
 $m = \log_2(n), \log_3(n), e^n$

## Tipos de órdenes de eficiencia

- Peor caso, umbral superior  $\Rightarrow O(\cdot)$
- Mejor caso, umbral inferior  $\Rightarrow \Omega(\cdot)$
- Orden exacto  $\Rightarrow$  Si  $O(f(n)) = \Omega(f(n)) \Rightarrow \Theta(f(n)) = O(f(n)) = \Omega(f(n))$