PREGUNTAS AA

- ## Definición asintótica de O

Dada una función $f : \mathbb{N} \to \mathbb{R}_0^+$, su orden O es el conjunto de las funciones acotadas superiormente, a partir de un cierto umbral, por múltiplos reales y positivos de f.

$$O(f) = \{t : \mathbb{N} \to \mathbb{R}_0^+ \mid \exists c \in \mathbb{R}^+ \ \exists n_0 \in \mathbb{N} \ \forall n \geqslant n_0 \ t(n) \leqslant cf(n)\}$$

Así, $t \in O(f)$ si, y solo si, $\exists c \in \mathbb{R}^+ \ \exists n_0 \in \mathbb{N} \ \forall n \geqslant n_0 \ t(n) \leqslant cf(n)$

Que la definición sea asintótica significa que solo importa lo que ocurre para valores de n suficientemente grandes.

- ## ¿Qué es un preorden? ¿Qué tipo de orden asintótico lo induce?

Es una relación de O (orden superior) binaria reflexiva y transitiva, definida por:

$$f \leqslant_O g \iff O(f) \subseteq O(g)$$

- ## Propiedades de O.

$$\begin{aligned}
f \in O(g) &\iff O(f) \subseteq O(g) \\
f \in O(g) \wedge g \in O(f) &\iff O(f) = O(g) \\
f \in O(g) \wedge g \notin O(f) &\iff O(f) \subset O(g)
\end{aligned}$$

$$O(cf) = O(f) \qquad (c \in \mathbb{R}^+)$$

$$O(f+g) = O(\text{máx}\{f,g\})$$

$$O\left(\sum_{i=0}^{k} c_i n^i\right) = O(n^k) \qquad (c_k \in \mathbb{R}^+)$$

- ## Jerarquía de Complejidad.

$$O(1) \leq O(\log n) \leq O(n) \leq O(n \cdot \log n) \leq O(n^2) \leq \cdots \leq O(n^a) \leq O(2^n) \leq O(n!)$$

Jesús Lagares Galán

- **Definición asintótica de omega.**

Dada una función $f : \mathbb{N} \to \mathbb{R}_0^+$, su orden Ω es el conjunto de las funciones acotadas inferiormente, a partir de un cierto umbral, por múltiplos reales y positivos de f.

$$\Omega(f) = \{t : \mathbb{N} \to \mathbb{R}_0^+ \mid \exists c \in \mathbb{R}^+ \ \exists n_0 \in \mathbb{N} \ \forall n \geqslant n_0 \ cf(n) \leqslant t(n)\}$$

Así, $t \in \Omega(f)$ si, y sólo si, $\exists c \in \mathbb{R}^+ \ \exists n_0 \in \mathbb{N} \ \forall n \geqslant n_0 \ cf(n) \leqslant t(n)$

- **¿Qué es la dualidad?**

La dualidad nos permite <<traspasar>> las propiedades de O (orden superior) a Omega (orden inferior) y viceversa.

- **Relación entre O y omega.**

$$O(f) = O(g) \iff \Omega(f) = \Omega(g)$$

- **Definición del orden exacto.**

Dada una función $f : \mathbb{N} \to \mathbb{R}_0^+$, su orden Θ es el conjunto de funciones acotadas (superior e inferiormente), a partir de un cierto umbral, por múltiplos reales y positivos de f.

$$\Theta(f) = \{t : \mathbb{N} \to \mathbb{R}_0^+ \mid \exists c_1, c_2 \in \mathbb{R}^+ \ \exists n_0 \in \mathbb{N} \ \forall n \geqslant n_0 \ c_1 f(n) \leqslant t(n) \leqslant c_2 f(n)\}$$

Así, $t \in \Theta(f)$ si, y solo si, $t \in \Omega(f)$ y $t \in O(f)$

- **¿Qué es una operación elemental? ¿Y una operación crítica?**

Una operación es elemental si se ejecuta en una máquina real en un tiempo acotada por una constante (t pertenece al orden exacto de 1).

Una operación es crítica si se ejecuta al menos tantas veces como cualquier otra del algoritmo. A la hora de elegir una operación crítica debemos tener en cuenta:

  - Que sea elemental.
  - Que se ejecute tantas veces como cualquier otra del algoritmo.
  - Que sea representativa.

- **Relación entre programa y algoritmo.**

  Un algoritmo es una secuencia finita de instrucciones capaz de computar la función, y un programa es la expresión de dicho algoritmo en un lenguaje de programación determinado.

- **Cualquier algoritmo de $O(n\log n)$ no siempre es mejor que cualquier algoritmo de $O(n2)$**

  VERDADERO. No podemos afirmar que SIEMPRE será mejor un algoritmo que otro porque entran más factores en juego (máquina, lenguaje de programación, etc).

- **$n\log n \in O(n2)$ y $n\log n \in \Omega(n2)$**

  FALSO

- **Independientemente del tamaño del problema, es siempre mejor un algoritmo de orden logarítmico que de orden lineal.**

  FALSO. Al igual que antes, dejando a un lado el tamaño del problema nos puede seguir afectando la arquitectura de la máquina, etc.

- **El número de veces que se realiza la operación crítica en el caso mejor es siempre inferior al del caso promedio.**

  FALSO. Puede ser igual al del caso promedio.

- $\sqrt{n2} <o\ n\log n <o\ n3+n <o\ 2n <o\ n! <o\ n3n$

  VERDADERO

- **$n3 \in O(2n)$ y $n3 \notin \Omega(2n)$**

  VERDADERO

- En el caso de ordenes parecidos ($n2$ *versus* n·log$n$), las constantes multiplicativas son decisivas tamaño del problema es lo bastante grande.

FALSO. Si el tamaño del problema es suficientemente grande, las constantes multiplicativas no importan. Para ilustrarlo, os pongo la siguiente tabla de los apuntes:

Nombre	$O(f(n))$	$n = 100$	$n = 200$
logarítmico	$\log n$	1 s	1,15 s
lineal	n	1 s	2 s
lineal logarítmico o cuasi-lineal	$n \log n$	1 s	2,30 s
cuadrático	n^2	1 s	4 s
cúbico	n^3	1 s	8 s
potencial	n^k	1 s	2^k s
exponencial	2^n	1 s	$1,27 \times 10^{30}$ s $>$ 4×10^{20} siglos

- Existen algoritmos de búsqueda en cualquier vector de coste O(log$n$)

FALSO. Creo que el algoritmo con ese coste es la búsqueda dicotómica, la cuál solo puede aplicarse en vectores ordenados, no en CUALQUIER VECTOR.

- Las constantes multiplicativas de los órdenes asintóticos carecen de importancia si el tamaño del problema es lo bastante pequeño.

FALSO. Si el tamaño es suficientemente pequeño, las constantes multiplicativas si importan, lo vemos en la siguiente tabla:

Nombre	$O(f(n))$	$n = 100$	$n = 200$
logarítmico	$\log n$	1 s	1,15 s
lineal	n	1 s	2 s
lineal logarítmico o cuasi-lineal	$n \log n$	1 s	2,30 s
cuadrático	n^2	1 s	4 s
cúbico	n^3	1 s	8 s
potencial	n^k	1 s	2^k s
exponencial	2^n	1 s	$1,27 \times 10^{30}$ s $>$ 4×10^{20} siglos