

Ejercicios resueltos de programación 3

Tema 3. Notación asintótica.

El *índice* de los ejercicios será el siguiente. Se observa que sólo hay cuestiones de exámenes (de los ejercicios cortos), por lo que se debería hacer hincapié en ellos:

1. Introducción teórica	3
2. Cuestiones de exámenes	6

Introducción teórica:

Previo a resolver los ejercicios pondremos un poco de teoría, que nos vendrá bien para luego hacer los ejercicios:

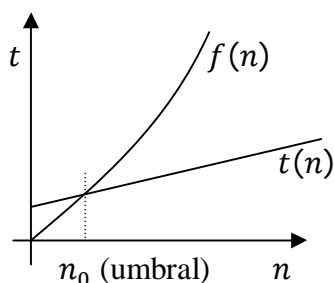
Empezaremos viendo las distintas notaciones, para "el orden de", cota inferior y orden exacto:

- Notación para el orden de (cota superior):

Es conveniente disponer de un símbolo matemático para representar *el orden de*. Sea $f: \mathbb{N} \rightarrow \mathbb{R}^{\geq 0}$ una función arbitraria de los números naturales en los reales no negativos. Le indicará mediante $O(f(n))$ el conjunto de todas las funciones $t: \mathbb{N} \rightarrow \mathbb{R}^{\geq 0}$ tales que $t(n) \leq c * f(n)$, para todo $n \geq n_0$ para una constante positiva c y un umbral entero n_0 . En otras palabras:

$$O(f(n)) \equiv \{t: \mathbb{N} \rightarrow \mathbb{R}^{\geq 0} | \exists c \in \mathbb{R}^+, n_0 \in \mathbb{N}, \forall n \geq n_0 | t(n) \leq c * f(n)\}$$

Gráficamente sería:



siendo:

n_0 : Cierta umbral del tamaño del problema.

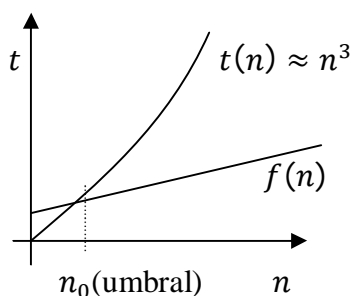
$f(n)$: Acota superiormente a la función $t(n)$.

- Notación para la cota inferior:

Matemáticamente, esto significa que existe una constante real positiva d y un umbral entero n_0 tal que $t(n) \geq d * f(n)$ siempre que $n \geq n_0$.

$$\Omega(f(n)) \equiv \{t: \mathbb{N} \rightarrow \mathbb{R}^{\geq 0} | \exists d \in \mathbb{R}^+, n_0 \in \mathbb{N}, \forall n \geq n_0 | t(n) \geq d * f(n)\}$$

Gráficamente sería:



$t(n) \in \Omega(f(n))$: Cota inferior.

$f(n) \in O(t(n))$: Cota superior.

- Notación para el orden exacto:

Diremos que $t(n)$ está en Theta de $f(n)$, o lo que es igual que $t(n)$ está en el orden exacto de $f(n)$ y lo denotamos $t(n) \in \theta(f(n))$, si $t(n)$ pertenece tanto a $O(f(n))$ como a $\Omega(f(n))$.

La definición formal de θ es:

$$\theta(f(n)) = O(f(n)) \cap \Omega(f(n)).$$

Por tanto,

$$\theta(f(n)) \equiv \{t: \mathbb{N} \rightarrow \mathbb{R}^{\geq 0} \mid \exists a, b \in \mathbb{R}^+, n_0 \in \mathbb{N}, \forall n \geq n_0 \mid \\ a * f(n) \leq t(n) \leq b * f(n)\}.$$

Decimos que el conjunto del orden exacto está acotado tanto inferior como superiormente por $f(n)$. Podemos probarlo tanto por la definición como por la regla del límite.

Para demostrar que una función dada no pertenece al orden de otra función $f(n)$ tendremos estas formas:

- Demostración por contradicción: Es la forma más sencilla. Consiste en demostrar la veracidad de una sentencia demostrando que si negación da lugar a una contradicción.
- La regla del umbral generalizado: Implica la existencia de una constante real y positiva c tal que $t(n) \leq c * f(n)$ para todos los $n \geq 1$ (tomaremos n_0 como 1, nos interesa más la definición dada por la regla del umbral sin generalizar).
- La regla del límite: Lo definiremos completamente tras analizar la cota superior y el coste exacto.

La primera y segunda manera no la usaremos por norma general, ya que no nos compensará. En cuanto a la última será la que usemos, de nuevo recordaremos la definición y lo que significa cada resultado:

La regla del límite: Nos permite comparar dos funciones en cuanto a la notación asintótica se refiere. Tendremos que calcular el siguiente límite:

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)}.$$

Se nos darán 3 resultados:

$$1. \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = c \in \mathbb{R} \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{lll} f(n) \in O(g(n)) & f(n) \in \Omega(g(n)) & f(n) \in \theta(g(n)) \\ g(n) \in O(f(n)) & g(n) \in \Omega(f(n)) & g(n) \in \theta(f(n)) \end{array} \right\}.$$

Estas funciones se comportan igual. Se diferencian en una constante multiplicativa.

$$2. \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \infty \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{lll} f(n) \notin O(g(n)) & f(n) \in \Omega(g(n)) & f(n) \notin \theta(g(n)) \\ g(n) \in O(f(n)) & g(n) \notin \Omega(f(n)) & g(n) \notin \theta(f(n)) \end{array} \right\}.$$

Por muy alta que sea la constante multiplicativa de $g(n)$ nunca superará a $f(n)$.

$$3. \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{lll} f(n) \in O(g(n)) & f(n) \notin \Omega(g(n)) & f(n) \notin \theta(g(n)) \\ g(n) \notin O(f(n)) & g(n) \in \Omega(f(n)) & g(n) \notin \theta(f(n)) \end{array} \right\}.$$

$g(n)$ crece más exponencialmente que $f(n)$. Sería su cota superior.

1ª parte. Cuestiones de exámenes:

Febrero 2002 -2ª (ejercicio 2)

Enunciado: Un algoritmo de coste $O(n^2)$ tarda 15 segundos en realizar un determinado procesamiento sobre un ordenador a 450 MHz. ¿Cuánto tiempo se tarda en realizar el mismo procesamiento con el mismo algoritmo en una máquina 3 veces más lenta?

Respuesta: Se nos plantea un problema en el que cambia la velocidad, que equivale a la implementación. En este caso, se divide por tres la velocidad, lo que implica que tarda aún más. Serían 15 seg. $\cdot 3 = 45$ seg.

Si en esta misma máquina cambiamos el tamaño del problema al doble tardaremos 4 veces más $((2 \cdot n)^2 = 4 \cdot n^2)$, si es triple serían 9 veces más, siguiendo el planteamiento anterior.

Diciembre 2003 (ejercicio 1)

Enunciado: ¿Qué significa que el tiempo de ejecución de un algoritmo está "en el orden exacto de $f(n)$ "? Demostrar que $T(n) = 5 \cdot 2^n + n^2$ está en el orden exacto de 2^n .

Respuesta:

Para la primera pregunta tendremos que poner la definición previamente escrita en la teoría o bien en el resumen del tema. No la pondremos, por estar en este mismo documento, en páginas anteriores.

Para la segunda pregunta emplearemos la regla del límite. En este caso, tomaremos $f(n) = 5 \cdot 2^n + n^2$ y $g(n) = 2^n$. Pasamos a resolver el límite, como sigue:

$$\lim_{n \rightarrow \infty} \frac{5 \cdot 2^n + n^2}{2^n}$$

Al resolverla llegamos a una indeterminación, por lo que aplicaremos el **teorema de L'Hôpital** tantas veces como sea necesario hasta llegar a una conclusión coincidiendo con cualquiera de los casos anteriores:

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{5 \cdot 2^n + n^2}{2^n} &= \lim_{n \rightarrow \infty} \frac{5 \cdot 2^n \cdot \log(2) + 2 \cdot n}{2^n \cdot \log(2)} = \lim_{n \rightarrow \infty} \frac{5 \cdot 2^n \cdot \log(2)^2 + 2}{2^n \cdot \log(2)^2} = \\ \lim_{n \rightarrow \infty} \frac{5 \cdot 2^n \cdot \log(2)^3}{2^n \cdot \log(2)^3} &= 5. \end{aligned}$$

Observamos que al final sería una constante numérica, por lo que estamos en el caso 1, concluyendo que $f(n) \in \theta(g(n))$ y, por tanto, queda demostrado.

Septiembre 2004 (ejercicio 1)

Enunciado: ¿Qué significa que el tiempo de ejecución de un algoritmo está "en el orden exacto de $f(n)$ "? Demostrar que $T(n) = n^3 + 9 \cdot n^2 \cdot \log(n)$ está en el orden exacto de n^3 .

Respuesta: La definición del orden exacto la hemos visto previamente. En cuanto a la demostración, se haría igual, empleando el **teorema de L'Hôpital**, por lo que evitamos realizar de nuevo el ejercicio por hacerlo previamente.

Septiembre 2005-reserva (ejercicio 2)

Enunciado: Demostrar formalmente si existe relación de pertenencia entre $f(n)$ y $O(g(n))$ y también entre $g(n)$ y $O(f(n))$ considerando $f(n) = 2^n$ y $g(n) = 2^{2*n}$

Respuesta: Se nos pide que veamos las relaciones entre la función y el orden de otra función dada, por tanto, tras ver la teoría previa, dada en el ejercicio anterior vemos que tenemos que usar la **regla del límite**:

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \lim_{n \rightarrow \infty} \frac{2^n}{2^{2*n}}.$$

En este caso, no podremos aplicar **L'Hôpital** por ser una exponencial, ya que la derivada quedaría igual:

$$\frac{da^n}{dn} = a^n.$$

Para resolverlo, tendremos que descomponer una de las dos funciones:

$$g(n) = 2^{2*n} = 2^n * 2^n.$$

Con esta nueva información pasamos a resolver el límite:

$$\lim_{n \rightarrow \infty} \frac{2^n}{2^n * 2^n} = \lim_{n \rightarrow \infty} \frac{1}{2^n} = 0.$$

Estaremos en el caso número 3 de los que vimos previamente, por tanto, deducimos que:

$$f(n) \in O(g(n)) \Rightarrow 2^n \in 2^{2*n}.$$

$$g(n) \notin O(f(n)) \Rightarrow 2^{2*n} \notin 2^n.$$

Septiembre 2005 (ejercicio 3)

Enunciado: Sea $T(n) = 4 * n^2 - 3 * n + 2$ el tiempo de ejecución de un algoritmo. Demuestra si es cierta o falsa cada una de las siguientes afirmaciones (0.5 puntos cada una):

- a) $T(n) \notin O(n^2 * \log(n))$
- b) $T(n) \notin O(n^3)$
- c) $T(n) \in \Omega(n * \log(n))$
- d) $T(n) \in O(n^2)$

Respuesta: Nuevamente usaremos la **regla del límite** y también **L'Hôpital**:

$$a) T(n) \notin O(n^2 * \log(n))$$

$$\lim_{n \rightarrow \infty} \frac{4*n^2 - 3*n + 2}{n^2 * \log(n)} = \lim_{n \rightarrow \infty} \frac{8*n - 3}{2*n * \log(n) + n^2 * \frac{1}{n}} = \lim_{n \rightarrow \infty} \frac{8}{2 * \log(n) + 3} = 0.$$

Pertenece al orden indicado, por tanto, la afirmación es **falsa**.

$$b) T(n) \notin O(n^3)$$

$$\lim_{n \rightarrow \infty} \frac{4*n^2 - 3*n + 2}{n^3} = \lim_{n \rightarrow \infty} \frac{8*n + 3}{3*n^2} = \lim_{n \rightarrow \infty} \frac{8}{6*n} = 0.$$

Por lo que $T(n) \in O(n^3)$, por lo que la afirmación es **falsa**.

c) $T(n) \in \Omega(n * \log(n))$

$$\lim_{n \rightarrow \infty} \frac{4*n^2 - 3*n + 2}{n * \log(n)} = \lim_{n \rightarrow \infty} \frac{8*n + 3}{\log(n) + n * \frac{1}{n}} = \lim_{n \rightarrow \infty} \frac{8}{\frac{1}{n}} = \infty.$$

Con lo que es **cierto**.

d) $T(n) \in O(n^2)$

$$\lim_{n \rightarrow \infty} \frac{4*n^2 - 3*n + 2}{n^2} = \lim_{n \rightarrow \infty} \frac{8*n - 3}{2*n} = \lim_{n \rightarrow \infty} \frac{8}{2} = 4 \in \mathbb{R}^+.$$

En este caso, al ser $T(n) \in \theta(n^2)$ se cumple también que $T(n) \in O(n^2)$, por lo que es **cierto**.

Febrero 2008-1ª (ejercicio 2)

Enunciado: ¿Cuáles de las siguientes respuestas son verdaderas y cuáles falsas? Demuestra tus respuestas.

- a) $n^2 \in O(n^3)$
- b) $n^2 \in \Omega(n^3)$
- c) $4 * n^3 - 3 * n + 2 \in \Omega(n * \log(n))$
- d) $n! \in \theta((2 * n + 1)!)$

Respuesta: Como en el ejercicio anterior emplearemos la misma técnica para resolverlo.

a) $n^2 \in O(n^3)$

Consideramos $f(n) = n^2$ y $g(n) = n^3$ y aplicamos el teorema del límite como sigue:

$$\lim_{n \rightarrow \infty} \frac{n^2}{n^3} = \lim_{n \rightarrow \infty} \frac{2*n}{3*n^2} = \lim_{n \rightarrow \infty} \frac{2}{6*n} = 0.$$

Esto significa que n^3 crece más rápidamente que n^2 , por tanto, es **cierto**.

b) $n^2 \in \Omega(n^3)$

En este caso, veríamos que con el resultado anterior, podremos afirmar que es **falso**.

c) $4 * n^3 - 3 * n + 2 \in \Omega(n * \log(n))$

$$\lim_{n \rightarrow \infty} \frac{4*n^3 - 3*n + 2}{n * \log(n)} = \lim_{n \rightarrow \infty} \frac{8*n - 3}{\log(n) + n * \frac{1}{n}} = \lim_{n \rightarrow \infty} \frac{8}{\frac{1}{n}} = \infty.$$

Es **cierto**, por ser $f(n) \in \Omega(n * \log(n))$.

d) $n! \in ((2 * n + 1)!)$

En este caso, no es posible usar **L'Hôpital** por ser un factorial, por lo que usaremos otro método. Lo resolveremos así:

$$\lim_{n \rightarrow \infty} \frac{n!}{(2 * n + 1)!} = \lim_{n \rightarrow \infty} \frac{(n) * (n - 1) * \dots * 1}{(2 * n + 1) * (2 * n) * (2 * n - 1) * \dots * (n) * (n - 1) * \dots * 1} = 0$$

Vemos que al desarrollar el factorial podemos eliminar muchos valores. En este caso, el resultado es 0, ya que $g(n)$ crecerá más rápidamente que $f(n)$.

Como resultado, es **falsa**, por ser $f(n) \notin \Omega(g(n))$ y, por tanto, $f(n) \notin \theta(g(n))$.