Análisis de Algoritmos y Estructuras de Datos

Tema 1: Órdenes asintóticos

Mª Teresa García Horcajadas
José Fidel Argudo Argudo
Antonio García Domínguez
Francisco Palomo Lozano

UCA Universidad de Cádiz

Versión 1.0

Índice

Repaso de conceptos básicos

Eficiencia

- Algoritmos y programas consumen recursos al ejecutarse
- Recursos en una máquina secuencial: tiempo y espacio
- A menor consumo de recursos, mayor eficiencia computacional
- Relacionamos eficiencia con tamaño de la entrada mediante funciones $\mathbb{N} \rightarrow \mathbb{R}_0^+$, donde $\mathbb{R}_0^+ = \mathbb{R}^+ \cup \{0\}$

Relación entre la eficiencia de programas y algoritmos

- Por el principio de invarianza, la eficiencia de todo programa para un mismo algoritmo solo varía en un factor constante
- Los órdenes asintóticos sirven para expresar la eficiencia sin tener en cuenta esos factores constantes
- Esto nos permite centrarnos en los algoritmos

Definición de O

Definición

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.

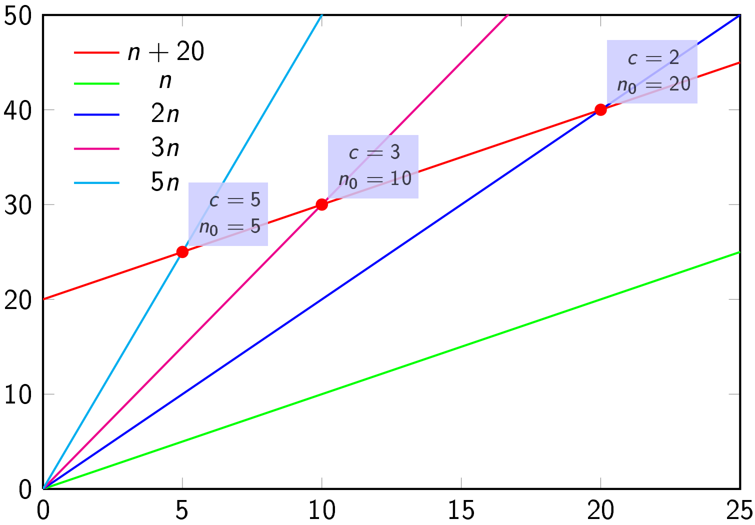
$$O(f) = \{t : \mathbb{N} \to \mathbb{R}_0^+ \mid \exists c \in \mathbb{R}^+ \ \exists n_0 \in \mathbb{N} \ \forall n \geqslant n_0 \ \ t(n) \leqslant cf(n)\}$$

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

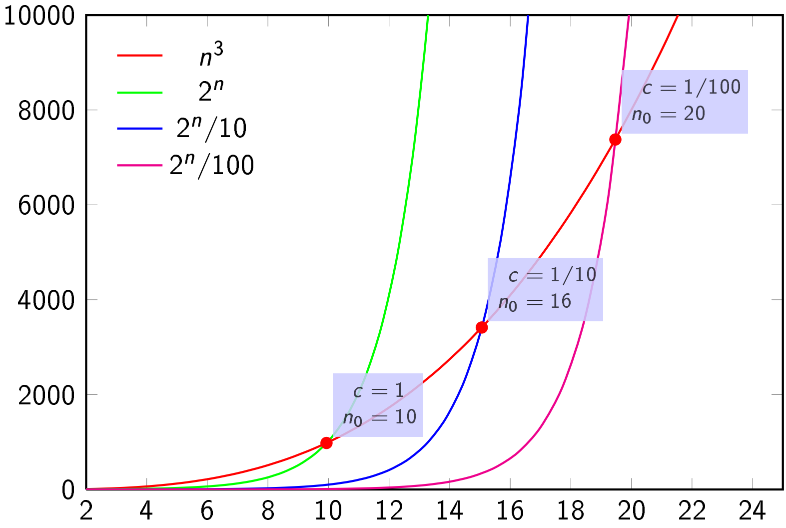
Nota

Esta definición es asintótica, ya que solo importa lo que ocurre para valores de n suficientemente grandes. Por lo tanto, se puede relajar cuando $n < n_0$ y permitir que f tome el valor 0, valores negativos o que incluso no esté definida.

Pertenencia de $n + 20$ a $O(n)$ con distintos c y n_0



Pertenencia de n^3 a $O(2^n)$ con distintos c y n_0



Propiedades de O (I)

Ordenación de funciones por órdenes

O induce un preorden $\leqslant_O$ (una relación binaria reflexiva y transitiva) sobre $\mathbb{N} \to \mathbb{R}_0^+$, definido por:

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

Este preorden no es total (existen elementos incomparables).

Pertenencia y contención

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

Propiedades de O (II)

Simplificación

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

$$O(f+g) = O(\max\{f,g\})$$

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

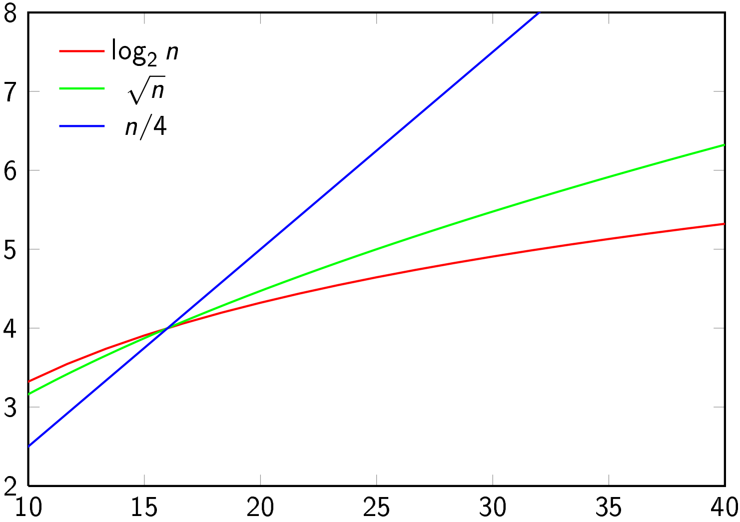
Comparación mediante límites (si existen)

$$\lim \frac{f(n)}{g(n)} = 0 \implies O(f) \subset O(g)$$

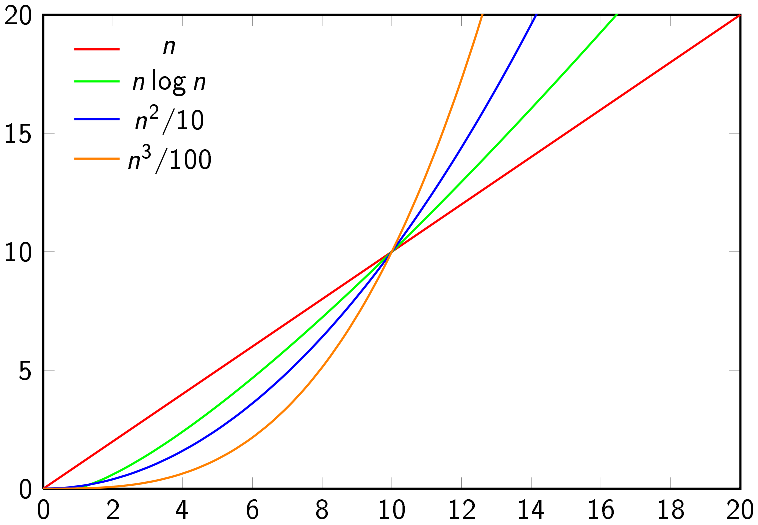
$$\lim \frac{f(n)}{g(n)} \in \mathbb{R}^+ \implies O(f) = O(g)$$

$$\lim \frac{f(n)}{g(n)} = \infty \implies O(g) \subset O(f)$$

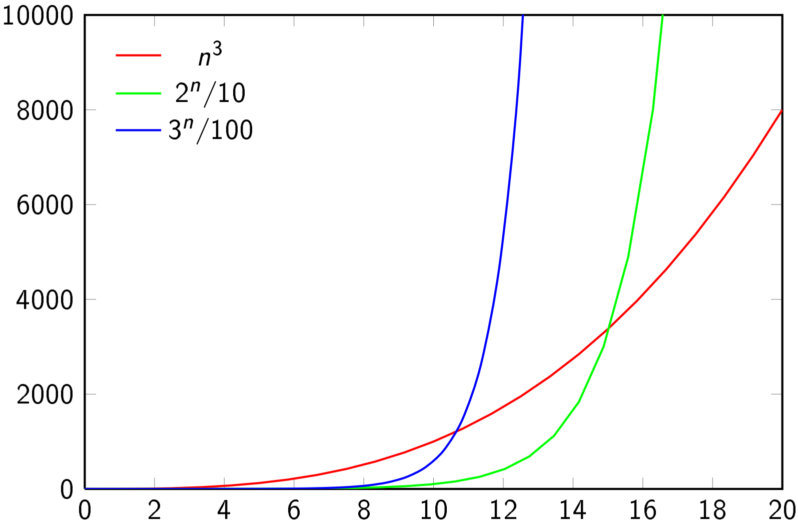
Jerarquía de complejidad: $\log_2 n <_O \sqrt{n} <_O n$



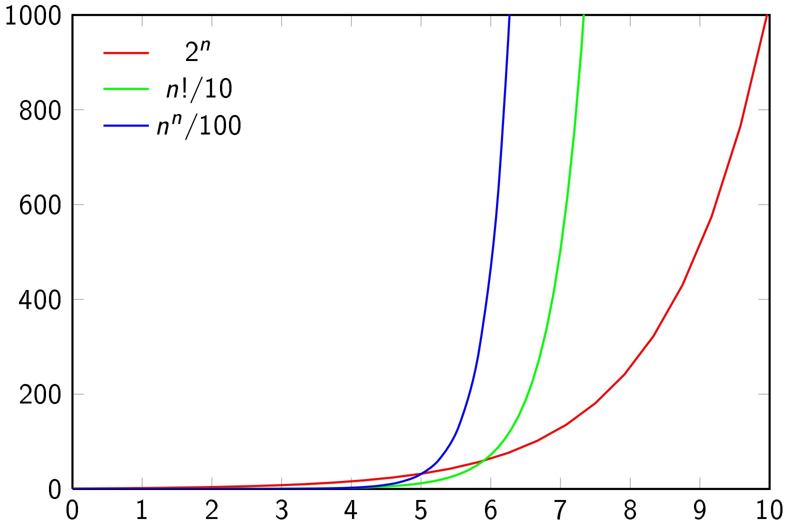
Jerarquía de complejidad: $n <_O n \log n <_O n^2 <_O n^3$



Jerarquía de complejidad: $n^3 <_O 2^n <_O 3^n$



Jerarquía de complejidad: $2^n <_O n! <_O n^n$



Jerarquía de complejidad

¿Qué ocurre si se multiplica el tiempo disponible?

Nombre	$O(f(n))$	$t = 1$ s	$t = 2$ s	$t = 10$ s
logarítmico	$\log n$	$n = 100$	$n = 10000$	$n = 10^{20}$
lineal	n	$n = 100$	$n = 200$	$n = 1000$
lineal logarítmico o cuasi-lineal	$n \log n$	$n = 100$	$n = 178$	$n = 702$
cuadrático	n^2	$n = 100$	$n = 141$	$n = 316$
cúbico	n^3	$n = 100$	$n = 126$	$n = 215$
potencial	n^k	$n = 100$	$n = 100 \cdot 2^{1/k}$	$n = 100 \cdot 10^{1/k}$
exponencial	2^n	$n = 100$	$n = 101$	$n = 103$

Jerarquía de complejidad

¿Qué ocurre si se dobla el tamaño de la entrada?

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

Definición de Ω

Definición

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, $\exists c \in \mathbb{R}^+ \; \exists n_0 \in \mathbb{N} \; \forall n \geqslant n_0 \;\; cf(n) \leqslant t(n)$

Propiedades de Ω

Dualidad

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

Permite «traspasar» las propiedades de O a Ω y viceversa.

Relación entre O y Ω

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

Definición de Θ

Definición

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.

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

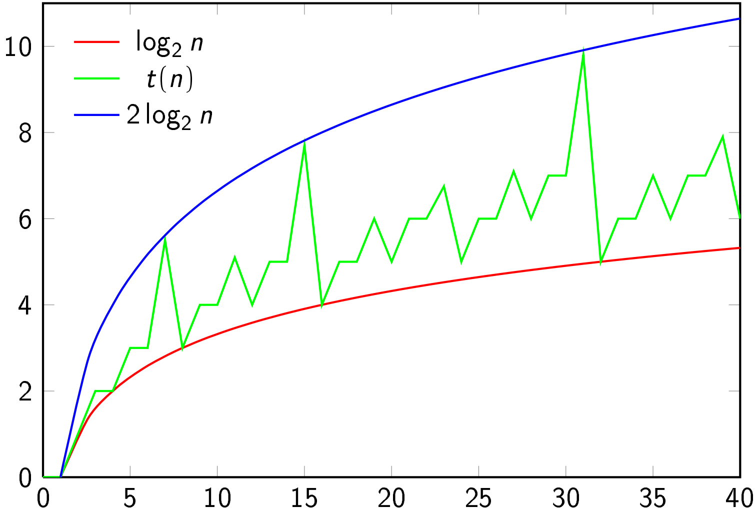
Así, $t \in \Theta(f)$ si, y solo si, $t \in \Omega(f)$ y $t \in O(f)$

Equivalencia entre órdenes de funciones

Θ induce una equivalencia $\equiv_\Theta$ (una relación binaria reflexiva, simétrica y transitiva) sobre $\mathbb{N} \to \mathbb{R}_0^+$, definida por:

$$f \equiv_\Theta g \iff \Theta(f) = \Theta(g)$$

Pertenencia a $\Theta(\log_2 n)$



Relación entre O, Ω y Θ

Propiedades

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

Las propiedades de O y Ω se traducen de manera sencilla a Θ.

Ejemplos

$$\Theta(cf) = \Theta(f)$$
$$\Theta(f+g) = \Theta(\max\{f,g\})$$
$$\lim \frac{f(n)}{g(n)} \in \mathbb{R}^+ \implies \Theta(f) = \Theta(g)$$

Simplemente por aplicación directa de la relación entre O y Θ.

Operaciones asintóticas

Definición

Dados $\Xi \in \{O, \Omega, \Theta\}$, $f, g : \mathbb{N} \to \mathbb{R}_0^+$ y un operador binario $\circ$, se define $\Xi(f) \circ \Xi(g)$ como el conjunto de las funciones que se obtienen aplicando $\circ$ a cada función de $\Xi(f)$ y de $\Xi(g)$.

$$\begin{aligned}\Xi(f) \circ \Xi(g) = \{t : \mathbb{N} \to \mathbb{R}_0^+ \mid \\ &\exists u \in \Xi(f)\ \exists v \in \Xi(g)\ \exists n_0 \in \mathbb{N} \\ &\forall n \geqslant n_0\ t(n) = u(n) \circ v(n)\}\end{aligned}$$

Suma y producto de órdenes

$$O(f) + O(g) = O(f + g) \qquad O(f) \cdot O(g) = O(f \cdot g)$$
$$\Omega(f) + \Omega(g) = \Omega(f + g) \qquad \Omega(f) \cdot \Omega(g) = \Omega(f \cdot g)$$
$$\Theta(f) + \Theta(g) = \Theta(f + g) \qquad \Theta(f) \cdot \Theta(g) = \Theta(f \cdot g)$$

Referencias

- Brassard, Gilles y Bratley, Paul.
  Algorítmica. Concepción y Análisis.
  Masson. 1990.
- Brassard, Gilles y Bratley, Paul.
  Fundamentos de Algoritmia.
  Prentice-Hall. 1997.
- Graham, Ronald L.; Knuth, Donald E. y Patashnik, Oren.
  Concrete Mathematics: A Foundation for Computer Science.
  Addison-Wesley. 1994. 2ª ed.