

# Análisis de algoritmos

Parte 2: Crecimiento de funciones

# Notación asintótica

- En este tema veremos algunos conjuntos de funciones con las que se suele representar la forma de crecimiento asintótico de un algoritmo
- Practicaremos su uso y calcularemos unas pocas de sus relaciones, pero sobre todo, veremos su significado gráfico

# ¿Qué es un conjunto?

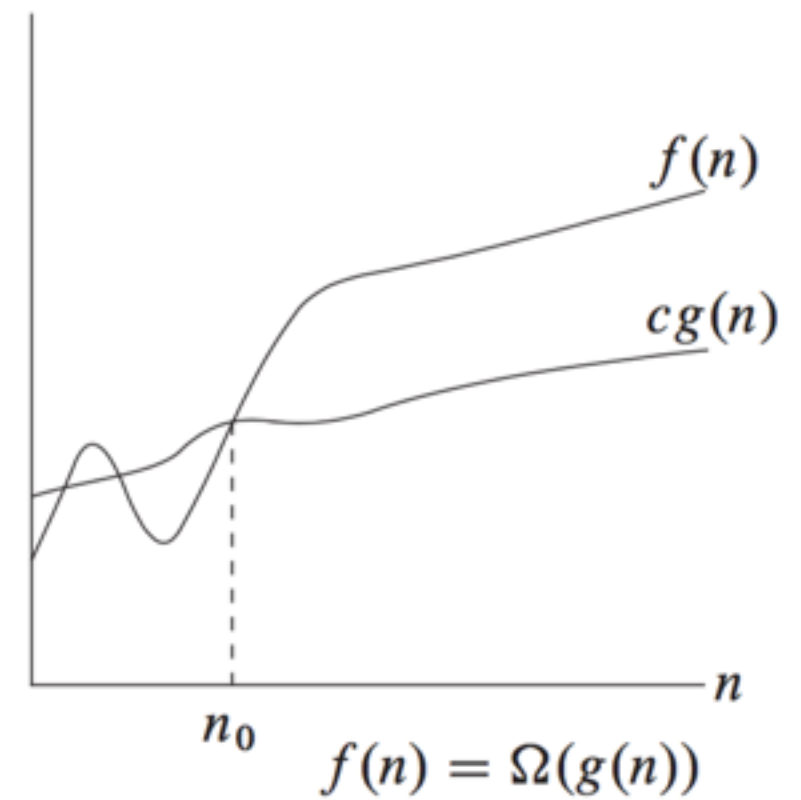
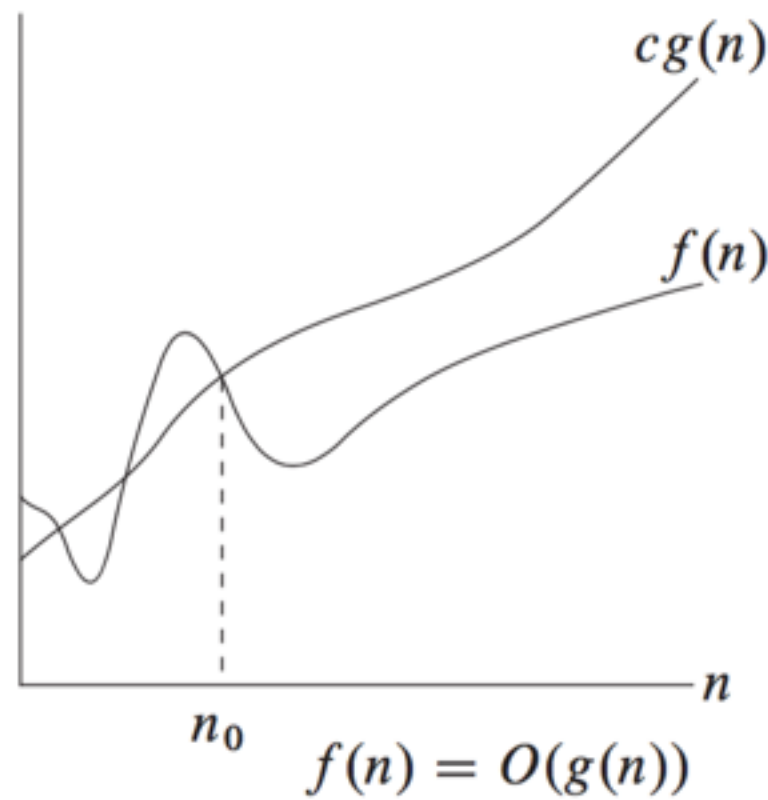
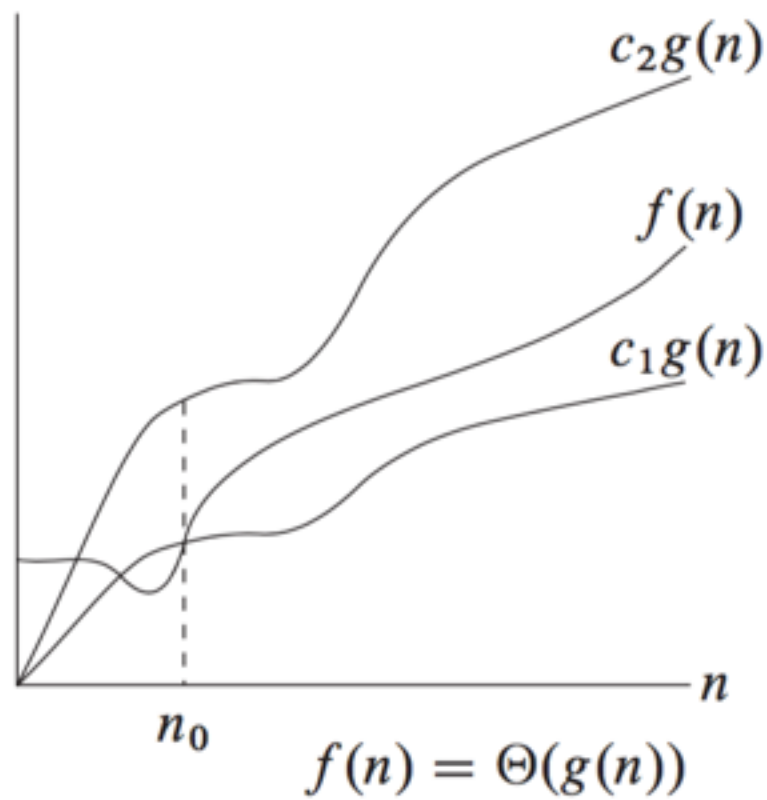
- Colección de elementos u objetos
- Con el nombre del conjunto en común
- No se admiten elementos repetidos

# Algunos conjuntos de funciones

- Rectas:  $\{Ax+B : A, B \in \mathbb{R}\}$
- Parábolas:  $\{Ax^2+Bx+C : A, B \in \mathbb{R}\}$
- Exponenciales:  $\{Ae^{Bx} : A, B \in \mathbb{R}\}$
- Logaritmos:  $\{A \ln(Bx) : A \in \mathbb{R}, B \in \mathbb{R}^+\}$
- Áreas de triángulos con la misma base:  
 $\{bx/2 : b \in \mathbb{R}^+\}$

- $O(g(n)) = \{f(n) : \text{existe } n_0 > 0 \text{ y } c > 0, \text{ tal que } 0 \leq f(n) \leq cg(n) \text{ para toda } n \geq n_0\}$
- $\Omega(g(n)) = \{f(n) : \text{existe } n_0 > 0 \text{ y } c > 0, \text{ tal que } 0 \leq cg(n) \leq f(n) \text{ para toda } n \geq n_0\}$
- $\Theta(g(n)) = \{f(n) : \text{existe } c_1 > 0, c_2 > 0, n_0 > 0, \text{ tal que } 0 \leq c_1g(n) \leq f(n) \leq c_2g(n) \text{ para toda } n \geq n_0\}$

# gráficamente



- El análisis asintótico se refiere a ver el comportamiento de las funciones para valores del parámetro  $n$  grandes
- Este análisis marca la tendencia de la función, o sea, lo que se puede esperar de ella
- Es el factor **más importante** y en general, **el más útil para analizar un algoritmo**, sin embargo no es el único y a veces las constantes que se “desprecian” son demasiado significativas para no considerarlas.
- Sin embargo, es más o menos fácil identificar cuando estas constantes van a ser significativas.

# Abusando de la notación

¿Cómo denotamos una función que está en estos conjuntos?

$f(n) \in O(g(n))$  se escribe  $f(n) = O(g(n))$

$f(n) \in \Omega(g(n))$  se escribe  $f(n) = \Omega(g(n))$

$f(n) \in \Theta(g(n))$  se escribe  $f(n) = \Theta(g(n))$



# El “orden” de una función

- Toda función se puede escribir  $f(x) = O(f(x))$
- En efecto, si  $f(x) \geq 0$ , tenemos que para  $c=1$  y  $n_0=1$  tenemos:

$$0 \leq f(x) \leq cf(x)$$

- De manera similar  $f(x) = \Omega(f(x))$  y  $f(x) = \Theta(f(x))$

- Así, si  $f(x) = \frac{1}{2}x + 3$
- entonces  $\frac{1}{2}x + 3 = O(\frac{1}{2}x + 3)$
- pero la importancia de esta notación asintótica es su poder para simplificar la notación de funciones
- Por ejemplo, podemos también escribir  $\frac{1}{2}x + 3 = O(x)$
- lo que demostraremos a continuación...

- Por definición, si queremos demostrar  

$$\frac{1}{2}x + 3 = O(x)$$
- entonces tenemos que mostrar que existen  

$$c > 0, n_0 > 0$$

- tal que

$$0 \leq f(x) \leq cg(x)$$

- donde

$$f(x) = \frac{1}{2}x + 3, g(x) = x$$

- Así tenemos

$$0 \leq \frac{1}{2}x + 3 \leq cx$$

- De la desigualdad derecha tenemos

$$\frac{1}{2}x + 3 \leq cx$$

- pasamos las  $x$  del lado mayor y lo demás del lado menor:

$$3 \leq cx - \frac{1}{2}x$$

- factorizamos  $x$

$$3 \leq (c - \frac{1}{2})x$$

- lo del paréntesis no puede ser cero ni negativo (pues el producto no podría ser mayor que 3), para poder dividir por lo de la izquierda, así escogemos un valor  $c > \frac{1}{2}$  por ejemplo  $c = 1$ , así:

$$x \geq \frac{3}{1 - \frac{1}{2}} = \frac{3}{\frac{1}{2}} = 6$$

- Con lo que el valor para  $n_0$  se escoge

$$n_0 = 6$$

- con lo que demostramos que efectivamente

$$\frac{1}{2}x + 3 = O(x)$$

- De la misma manera podemos demostrar que

$$\frac{1}{2}x + 3 = \Omega(x)$$

- así, queremos demostrar que existen

$$c > 0, n_0 > 0$$

- tales que

$$0 \leq cx \leq \frac{1}{2}x + 3$$

- Con la desigualdad derecha observamos que

$$\begin{aligned} cx &\leq \frac{1}{2}x + 3, \\ -3 &\leq \frac{1}{2}x - cx, \\ -3 &\leq \left(\frac{1}{2} - c\right)x \end{aligned}$$

- Lo del paréntesis no puede ser cero ni negativo, o al pasar dividiendo, la  $x$  estará del lado menor, y queremos que esté del lado mayor. Además la  $c$  debe ser positiva, así  $0 < c < 1/2$ , tomemos  $c = 1/4$ .

- Con lo que tenemos que  $x \geq -3/(1-1/4) = -12$
- Si  $n_0 = 6$ , todo lo anterior se cumplirá para  $x \geq n_0$
- con lo que efectivamente  $\frac{1}{2}x + 3 = \Omega(x)$
- Con estas dos pruebas y el siguiente teorema tendremos que  $\frac{1}{2}x + 3 = \Theta(x)$
- En efecto, el Teorema 1 relaciona  $O$ ,  $\Omega$  y  $\Theta$

$$f(x) = O(g(x)), f(x) = \Omega(g(x)) \iff f(x) = \Theta(g(x))$$

- Probemos primero que  

$$f(x) = O(g(x)), f(x) = \Omega(g(x)) \implies f(x) = \Theta(g(n))$$
- Así sabemos que existen constantes  

$$n_0 > 0, m_0 > 0, c_1 > 0, c_2 > 0$$
- tales que  

$$0 \leq c_1 g(n) \leq f(n)$$
- y  

$$0 \leq f(n) \leq c_2 g(n)$$
- tomemos  

$$k_0 = \max\{n_0, m_0\}$$
- con lo que para toda  $x > k_0$  se tiene  

$$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$$

- Probemos ahora que  

$$f(x) = \Theta(g(n)) \implies f(x) = O(g(x)), f(x) = \Omega(g(x))$$
- así, sabemos que existen  

$$n_0 > 0, c_1 > 0, c_2 > 0$$
- tales que para toda  $x > n_0$  se tiene  

$$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$$
- para toda  $x > n_0$ , por lo que
- de la 1a y 2a desigualdad (desde la izquierda) tenemos que  

$$f(x) = \Omega(g(x))$$
- para de la 1a y la 3a desigualdad tenemos que  

$$f(x) = O(g(x))$$





- Antes de ver más ejemplos y resultados, veamos la interpretación de estos conjuntos de funciones, en adelante simplemente las llamaremos “notación asintótica” en lo que se ha visto hasta ahora de la complejidad espacio-temporal de algoritmos, en particular veremos los ejemplos de inserción y mezclas.

# Complejidad temporal del algoritmo de inserción

- En el algoritmo de ordenamiento por inserción vemos que tiene el ciclo (que empieza en la línea 1) que controla la variable  $j$  y dentro tiene un ciclo (que empieza en la línea 4), que controla la variable  $i$ .
- La variable  $j$  toma valores desde 1 hasta  $\text{len}(A)-1$ , por lo que produce  $n-1$  iteraciones.
- A su vez, el ciclo de la variable  $i$  produce entre 1 y  $j-1$  iteraciones.
- El **mejor caso** (la lista está ordenada) es cuando el ciclo de  $i$  produce 1 iteración por cada iteración de  $j$ , o sea  **$n-1$  iteraciones en total**.
- Mientras que en el peor caso es cuando se producen  $j-1$  iteraciones por cada iteración de  $j$ .
- Como la cantidad total de iteraciones se puede calcular con la suma de los enteros desde 1 hasta  $n-1$ , esta cuenta nos da  **$n(n-1)/2$  iteraciones en total** en el **peor caso** (cuando la lista está ordenada al revés).

`inserción(A):`

1. Para  $j$  de 1 a  $\text{len}(A)-1$
2.      $\text{pivote} = A[j]$  # carta a acomodar
3.      $i = j - 1$  # posición a la izquierda del pivote
4.     Mientras  $i \geq 0$  y  $A[i] > \text{pivote}$
5.          $A[i+1] = A[i]$
6.          $i = i - 1$
7.      $A[i + 1] = \text{pivote}$

- relación de  $O$  y  $\omega$  con esto

- Si  $a_n \neq 0$

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0 = \Theta(x^n)$$

# Para cotas más justas existen...

- $o(g(n)) = \{f(n) : \text{para toda } c > 0, \text{ existe } n_0 > 0 \text{ tal que } 0 \leq f(n) \leq cg(n) \text{ para toda } n \geq n_0\}$
- $\omega(g(n)) = \{f(n) : \text{para toda } c > 0, \text{ existe } n_0 > 0 \text{ y, tal que } 0 \leq cg(n) \leq f(n) \text{ para toda } n \geq n_0\}$
- estas denotan “cotas justas”