Algoritmia y Complejidad

Titulación: Grado en Ingeniería Informática A

Curso: 2023-2024

Trabajo: Órdenes de Magnitud

Autor: Ana Martín Conejo

LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE MÁLAGA

UNIVERSIDAD DE MÁLAGA

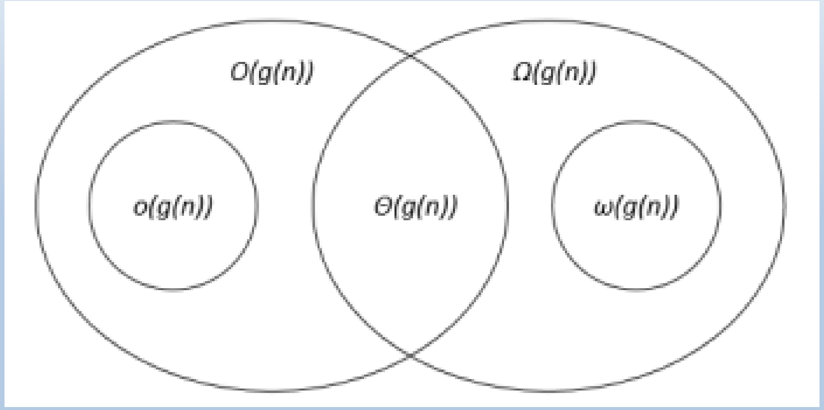
Enunciado

Se pide, para cada uno de los 5 órdenes de magnitud:

- Decir con palabras (adjetivos, adverbios), lo que significa.
- Proponer ejemplos sencillos de funciones:
  - de pertenencia a cada orden
  - de no pertenencia a cada orden
  - situaciones extremas (funciones que serían la frontera del orden de magnitud)

Órdenes de Magnitud a estudiar

$f(n) \in O(g(n))$	$\exists\, c > 0,$	$\exists\, n_0 > 0 : \forall\, n \geq n_0,$	$0 \leq \mathbf{f(n)} \leq c\, g(n)$	**$\lim (f(n)/g(n)) < \infty$**
$f(n) \in \Omega(g(n))$	$\exists\, c > 0,$	$\exists\, n_0 > 0 : \forall\, n \geq n_0,$	$0 \leq c\, g(n) \leq \mathbf{f(n)}$	**$\lim (f(n)/g(n)) > 0$**
$f(n) \in \Theta(g(n))$	$\exists\, c1, c2 > 0,$	$\exists\, n_0 > 0 : \forall\, n \geq n_0,$	$0 \leq c1\, g(n) \leq \mathbf{f(n)} \leq c2\, g(n)$	**$\lim (f(n)/g(n) = a$** **$a \neq 0, a \neq \infty$**
$f(n) \in o(g(n))$	$\forall\, c > 0,$	$\exists\, n_0 > 0 : \forall\, n \geq n_0,$	$0 \leq \mathbf{f(n)} < c\, g(n)$	**$\lim (f(n)/g(n)) = 0$**
$f(n) \in \omega(g(n))$	$\forall\, c > 0,$	$\exists\, n_0 > 0 : \forall\, n \geq n_0,$	$0 \leq c\, g(n) < \mathbf{f(n)}$	**$\lim (f(n)/g(n)) = \infty$**



Significado Informal

Orden de magnitud	Informalmente significa
O(g(n))	Es el conjunto de funciones acotadas superiormente por un múltiplo de g, se utiliza para probar que la complejidad de un algoritmo como muy mal se comportará como la función g tomada como referencia.
Ω(g(n))	Es el conjunto de funciones acotadas inferiormente por un múltiplo de g, se utiliza para probar que la función de complejidad de un algoritmo en el mejor caso se va a comportar como la función g, que se toma como referencia.
Θ(g(n))	Es el conjunto de funciones con el mismo orden de crecimiento que g, se utiliza para probar que la complejidad de un algoritmo es igual asintóticamente que g.
o(g(n))	Es el conjunto de funciones cuyo crecimiento es estrictamente menor que el de g cuando n tiende a infinito, esto implica que a medida que el tamaño de la entrada aumenta indefinidamente, la función crecerá mucho más lento que g.
ω(g(n))	Es el conjunto de funciones cuyo crecimiento es estrictamente mayor que g, luego van a crecer mucho más rápidos que este al contrario que en el caso anterior.

Ejemplos que pertenecen

Orden de magnitud	Pertenece
O(g(n))	$0.5n\ (n - 1) \in O(n^2)$
Ω(g(n))	$n^3 \in \Omega(n^2)$
Θ(g(n))	$100n + 23 \in \Theta(n)$
o(g(n))	$\log(n) \in o(n)$
ω(g(n))	$n^3 \in \omega\ (n)$

Ejemplos que no pertenecen

Orden de magnitud	No pertenece
O(g(n))	$n\,4 + n + 6 \notin O(n^2)$
Ω(g(n))	$n \notin \Omega(n^2)$
Θ(g(n))	$0.01n \notin \Theta(n^2)$
o(g(n))	$n^2 \notin o(n)$
ω(g(n))	$\log(n) \notin \omega(n)$

Ejemplos de casos extremos

Orden de magnitud	Casos extremos
O(g(n))	1 ∈ O(f) (siendo f cualquier función no constante)
Ω(g(n))	2^n∈ Ω (n^k) (k: grado del polinomio)
Θ(g(n))	n^k∈Θ(n^k) (k: grado del polinomio)
o(g(n))	1∈o(log(n)) ; log(log(n)) ∈o(n^2)
ω(g(n))	2^n∈ ω (n) ; n! ∈ ω (2^n)