

Estructuras de Datos y Algoritmos: Notación Asintótica

Yerko Ortiz

Objetivo: El objetivo de la segunda clase es estudiar una serie de funciones matemáticas bien conocidas y definir la notación Big-oh, Theta y Omega.

Índice

1. Introducción	1
2. Notación Asintótica	1
3. Crecimiento de funciones	2
4. Notación Big O	2
5. Notación Theta	2
6. Notación Omega	2

1. Introducción

Para poder medir la eficiencia de un algoritmo y así mismo poder comparar qué algoritmo podría ser mejor para un determinado problema, es necesario tener un método que describa la eficiencia del algoritmo en términos del tamaño de entrada del problema.

2. Notación Asintótica

La notación asintótica describe el comportamiento de una función matemática. En el contexto del curso, la notación asintótica caracteriza el tiempo de ejecución o espacio de memoria de un algoritmo mediante una función matemática.

Las ventajas de describir la eficiencia de un algoritmo con notación asintótica son:

- Describe la eficiencia en función de un tamaño variable de entrada.
- Caracteriza el uso de recursos del algoritmo mediante una función matemática sin constantes ni términos de menor orden.
- Es independiente de las especificaciones de la maquina que ejecuta el algoritmo.

3. Crecimiento de funciones

Puesto que el tamaño de entrada un algoritmo es arbitrario, el tiempo de ejecución o espacio de memoria de un algoritmo puede ser caracterizado mediante una función matemática que considere el tamaño de entrada.

Input	Constante	Logaritmo	Lineal	Linearítmica	Cuadrática
$f(N)$	1	$\lg N$	N	$N \lg N$	N^2
f(1000)	1	10	1000	10000	1000000
Tiempo	1 ns	10 ns	1 μ s	10 μ s	1 ms
Memoria	1 B	10 B	1KB	10KB	10MB

4. Notación Big O

$\mathcal{O}(g(N)) = \{f(N) : \text{existen constantes positivas } c \text{ y } N_0 \text{ de forma que } 0 \leq f(N) \leq cg(N), \text{ para todo } N \geq N_0\}$

- Big O caracteriza el peor caso de un algoritmo.

5. Notación Theta

$\Theta(g(N)) = \{f(N) : \text{existen constantes positivas } c_1, c_2 \text{ y } N_0 \text{ de forma que } 0 \leq c_1g(N) \leq f(N) \leq c_2g(N), \text{ para todo } N \geq N_0\}$

- Theta caracteriza el caso promedio de un algoritmo.

6. Notación Omega

$\Omega(g(N)) = \{f(N) : \text{existen constantes positivas } c \text{ y } N_0 \text{ de forma que } 0 \leq cg(N) \leq f(N), \text{ para todo } N \geq N_0\}$

- Omega caracteriza el mejor caso de un algoritmo.