Data Structures & Algorithms – 1 (DSA 1)

Topic: Asymptotic Notation

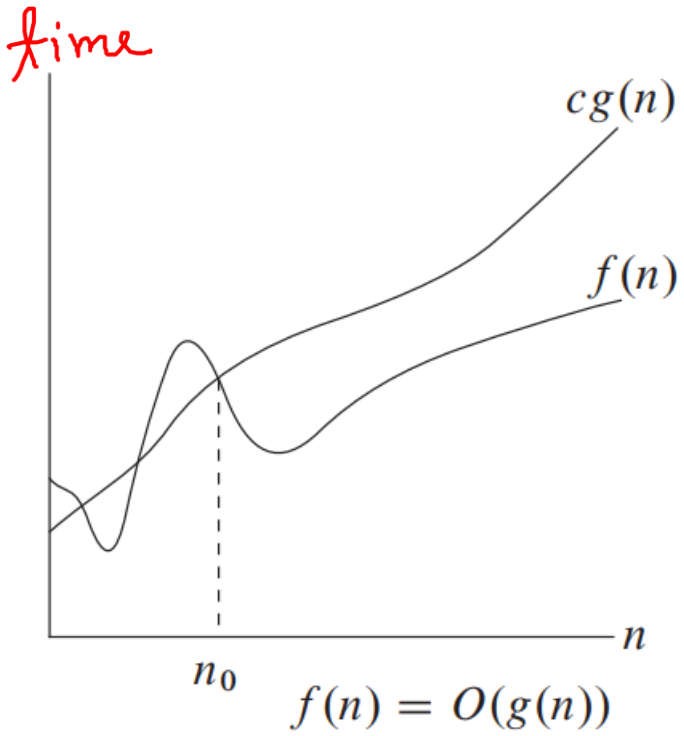
Learn With Mahfuz

Asymptotic Notation

1. Big-Oh (O) → Upper Bound
2. Big-omega (Ω) → Lower Bound
3. Theta (Θ) → Tight Bound

4. o-notation (Little-oh)
5. ω-notation (Little-ω)

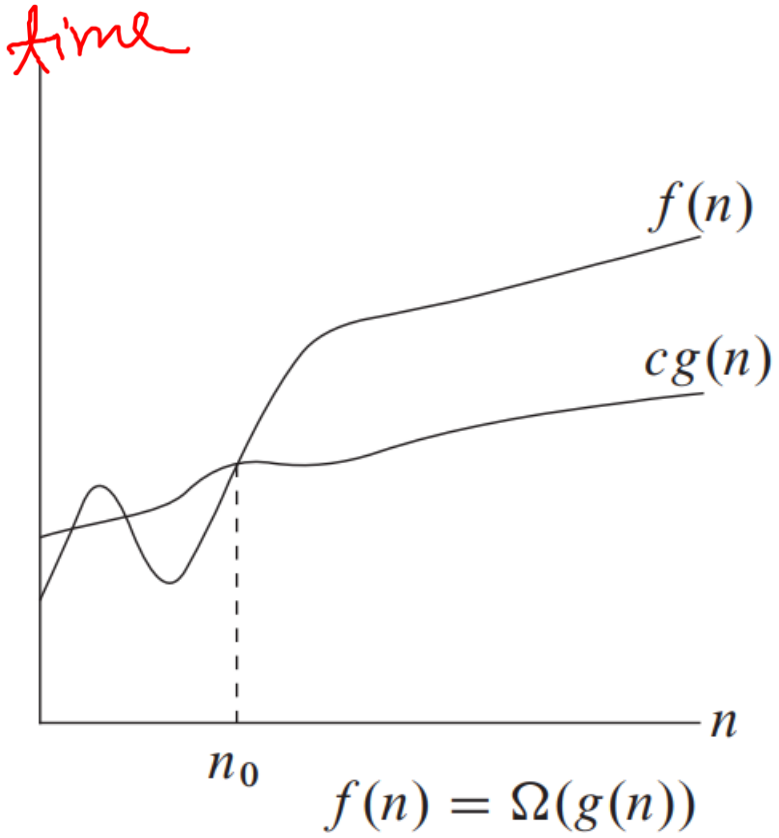
O-notation (big-oh)



$f(n) = O(g(n))$

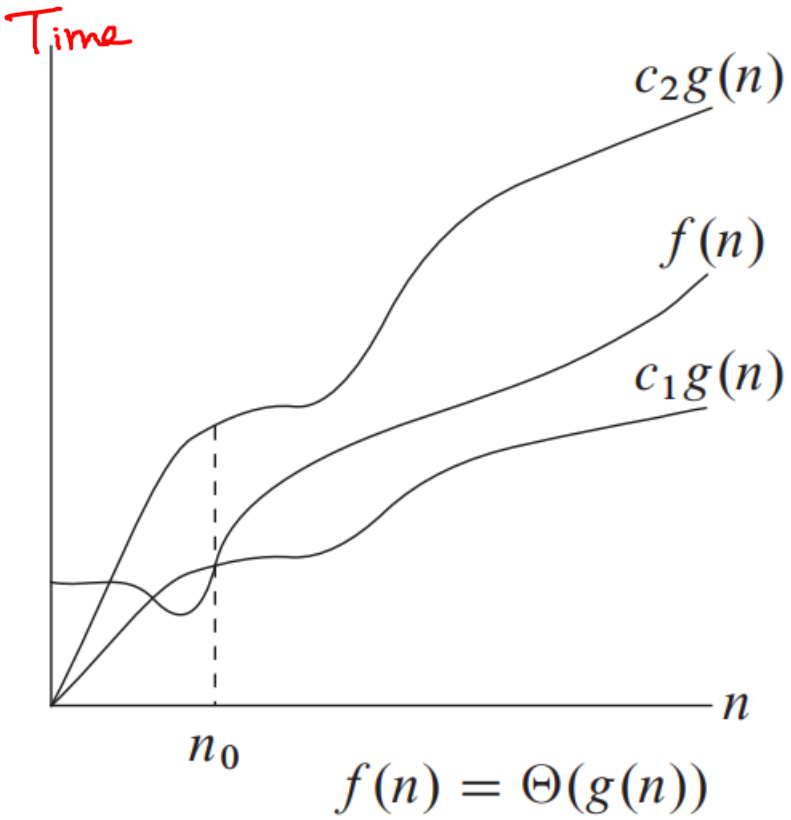
$$O(g(n)) = \{f(n) : \text{there exist positive constants } c \text{ and } n_0 \text{ such that } 0 \le f(n) \le cg(n) \text{ for all } n \ge n_0\} .$$

Ω-notation (big-omega)



$$\Omega(g(n)) = \{f(n) : \text{there exist positive constants } c \text{ and } n_0 \text{ such that } 0 \le cg(n) \le f(n) \text{ for all } n \ge n_0\} .$$

Θ-notation (theta)

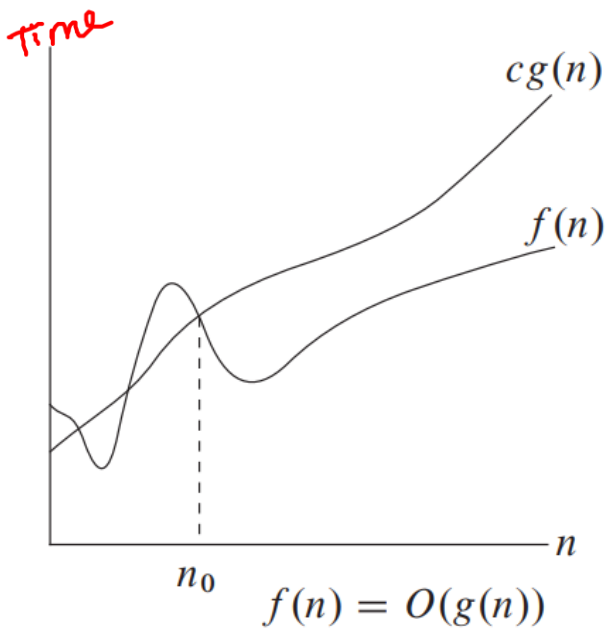


$f(n) = \Theta(g(n))$

$\Theta(g(n)) = \{f(n)$: there exist positive constants c_1, c_2, and n_0 such that $0 \le c_1 g(n) \le f(n) \le c_2 g(n)$ for all $n \ge n_0\}$.[1]

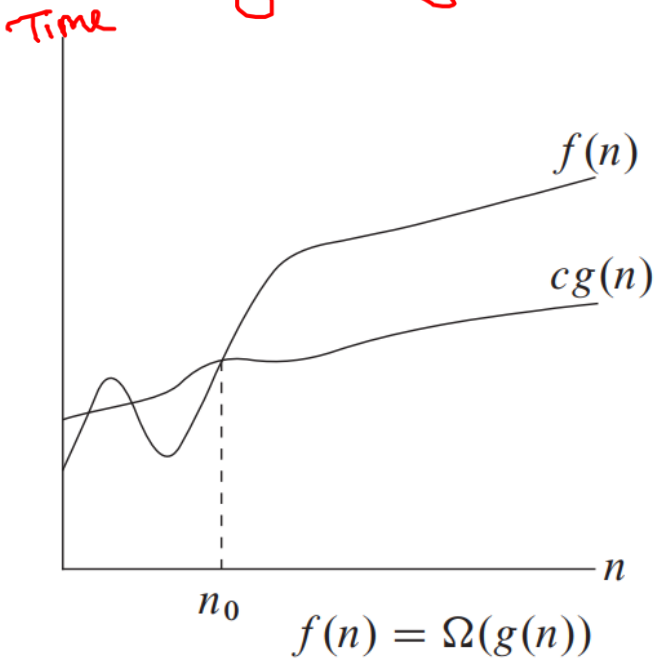
big-oh, big-omega, theta

Big-oh



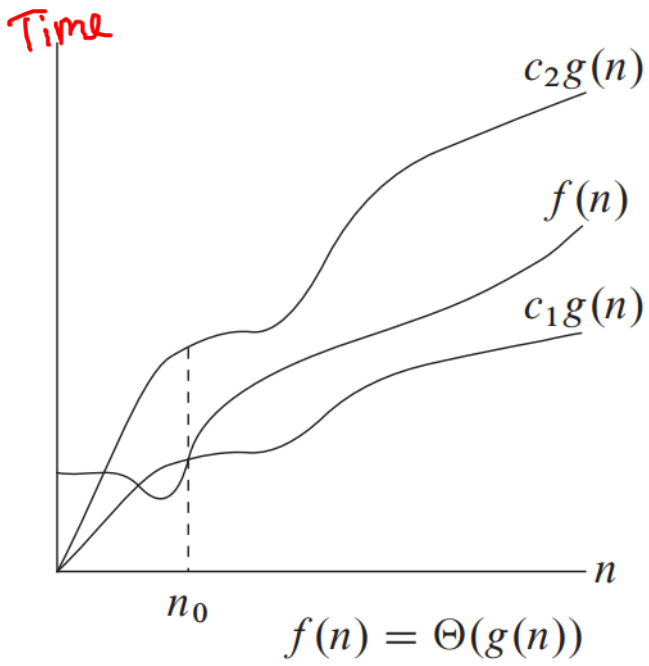
$f(n) \leq c \cdot g(n)$

Big-Omega



$f(n) \geq c \cdot g(n)$

Theta



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

* $c, c_1, c_2, n_0 \rightarrow +ve$
$n \geq n_0$

How to remember?

Big-Oh (Upper Bound): $f(n) \leq c \cdot g(n)$

Big-Omega (Lower Bound): $f(n) \geq c \cdot g(n)$

Theta (Tight Bound): $c_1 g(n) \leq f(n) \leq c_2 g(n)$

$c, c_1, c_2, n_0 \rightarrow +ve$

$n \geq n_0$

Little-oh : $f(n) < c \cdot g(n)$

Little-omega : $f(n) > c \cdot g(n)$

More efficient (less time)

Function	Descriptor	Big-Oh
c	Constant	$O(1)$
$\log n$	Logarithmic	$O(\log n)$
n	Linear	$O(n)$
$n \log n$	$n \log n$	$O(n \log n)$
n^2	Quadratic	$O(n^2)$
n^3	Cubic	$O(n^3)$
n^k	Polynomial	$O(n^k)$
2^n	Exponential	$O(2^n)$
$n!$	Factorial	$O(n!)$

Slower (more time)

Click here to go to the **GitHub repository**

Learn With Mahfuz
Let's Code Together

Click here to see this video!

THANK YOU!

Learn With Mahfuz
Let's Code Together