

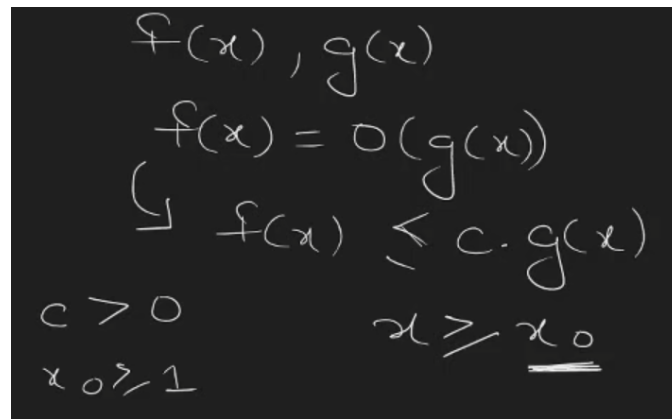
# 04\_Asymptotic Notations

Running time of an Algorithm

Time and space complexity of an algorithm

- Big O notation
- Omega Notation
- Theta Notation

1. Big O notation: WORST CASE SCENARIO(upper bound)



Handwritten mathematical definition of Big O notation on a blackboard:

$$f(x), g(x)$$
$$f(x) = O(g(x))$$
$$\Downarrow f(x) \leq c \cdot g(x)$$
$$c > 0 \quad x \geq \underline{x_0}$$
$$x_0 \geq 1$$

# Complexity

## Increasing order of Complexity

- 1) Constant time Complexity  $\rightarrow O(1)$
- 2) Logarithmic time Complexity  $\rightarrow O(\log n) \rightarrow$  Binary Search
- 3) Linear time Complexity  $\rightarrow O(n) \rightarrow$  Linear Search
- 4) Quadratic time Complexity  $\rightarrow O(n^2) \rightarrow$  Sorting
- 5) Cubic time Complexity  $\rightarrow O(n^3) \rightarrow$  Matrix Multiplication
- 6) Polynomial time Complexity  $\rightarrow O(n^c)$  where  $c$  is const.
- 7) Exponential time Complexity  $\rightarrow O(c^n)$  where  $c > 1$

Best

Worst  
Worst

Note :  $2^n < n! < n^n$  we can write  $2^n \rightarrow O(n!)$   
 $n! \rightarrow O(n^n)$

## 2. Omega Notation: Best Case Scenario(Lower Bound)

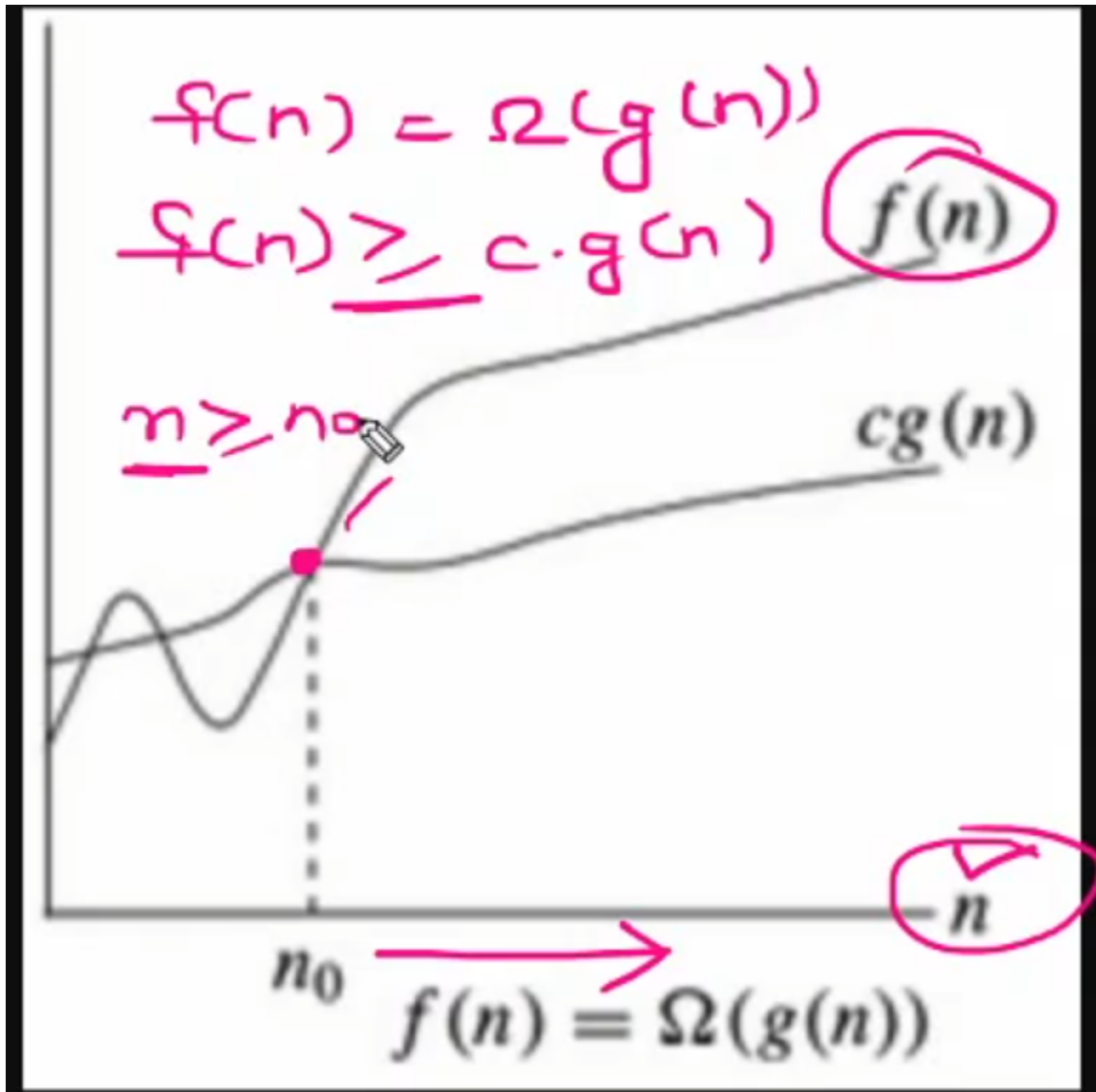
$f(n) = \Omega(g(n))$   $\Omega \rightarrow \Omega$   
 $\hookrightarrow$  Lower Bound  
 if and only  $\hookrightarrow$  Best case scenario  
 if  $f(n) \geq c \cdot g(n)$   $\text{Big O}$   $(f(n) \leq c \cdot g(n))$  True -  $f(n) = \Omega(g(n))$   
 $n \geq n_0$   $\checkmark$  (Graph)  $n_0 \geq 1$   $c > 0 \checkmark$   $n \geq c \cdot 5n$

1)  $f(n) = n$   
 $g(n) = 5n$   
 $f(n) = \Omega(g(n))$   
 $f(n) \geq c \cdot g(n)$   
 $n \geq c \cdot 5n$   $c = \frac{1}{5} > 0$

2)  $f(n) = 5n$ ,  $g(n) = n$   
 $f(n) = \Omega(g(n))$   
 $f(n) \geq c \cdot g(n)$   
 $5n \geq c \cdot n$   $c = 5 > 0$

3)  $f(n) = n^2$   
 $g(n) = n^2 + n + 10$   
 $f(n) = \Omega(g(n))$   $c = \frac{1}{2} > 0$   
 $f(n) \geq c \cdot g(n)$   
 $n^2 \geq c(n^2 + n + 10)$

$f(n) = \Omega(g(n))$   
 4)  $f(n) = n$   
 $g(n) = n^2$   
 $f(n) = \Omega(g(n))$   $\times$   
 $f(n) \geq c \cdot g(n)$   
 $n \geq c \cdot n^2$   
Not  
constant  $c = \frac{1}{n}$



3. Theta Notation: Satisfies both, omega and theta notation

# Theta Notation ( $\Theta$ )

→ satisfies both Big O and Omega ✓

$$1) \quad f(n) = n \\ g(n) = 5n$$

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

Big O  
→  $f(n) = O(g(n))$

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

$$n \leq c_1 \cdot 5n$$

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

$$n \geq n_0$$

$$n_0 \geq 1$$

$$c > 0$$

$$f(n) \geq c_2 \cdot g(n)$$

$$n \geq n_0$$

$$n_0 \geq 1$$

$$c > 0$$

$$c_1 = 1$$

## Omega

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

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

$$n \geq c_2 \cdot 5n$$

$$c_2 = 1/5 > 0$$

$$2) \quad f(n) = n, \quad g(n) = n$$

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

$$1) \quad f(n) = O(g(n))$$

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

$$n \leq c \cdot n$$

$$c = 1$$

$$2) \quad f(n) = \Omega(g(n))$$

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

$$n \geq c \cdot n$$

$$c = 1$$