→ **Asymptotic Notations:** The main idea of asymptotic analysis is to have a measure of efficiency of algorithms that doesn't depend on machine specific constants.

- Asymptotic notations are the mathematical notations used to describe the running time (Time complexity) of an algorithm.

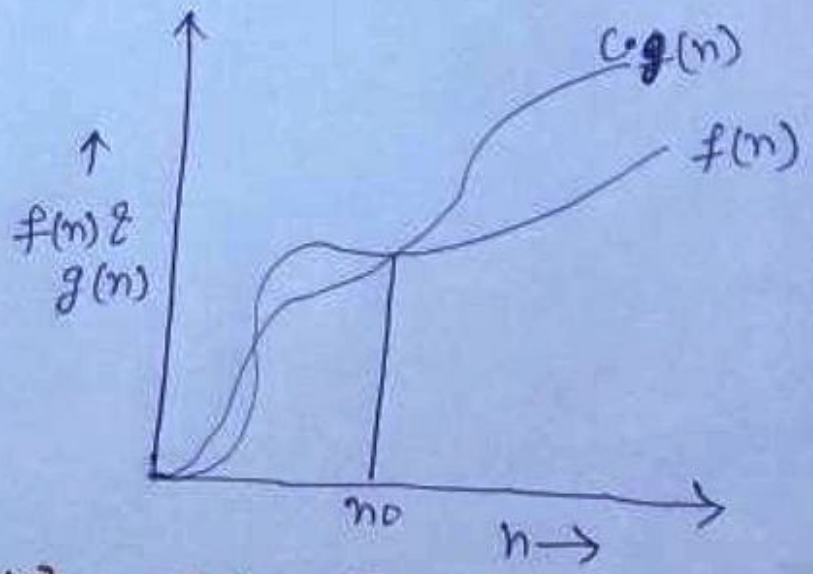
1. **Big O Notation:**
   - Worst case
   - Upper bound

Let $f(n)$ & $g(n)$ are two +ve functions.

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

iff

$f(n) \leq c \cdot g(n)$, $\forall n$, $n \geq n_0$ where c is a constant and value of $c > 0$, n_0 is constant and value of $n_0 \geq 1$.



P-1 $g(n) = n^2$, $f(n) = n^2 + n + 10$

P-2 $f(n) = n + 10$, $g(n) = n - 10$

P-3 $f(n) = n^2$, $g(n) = n$

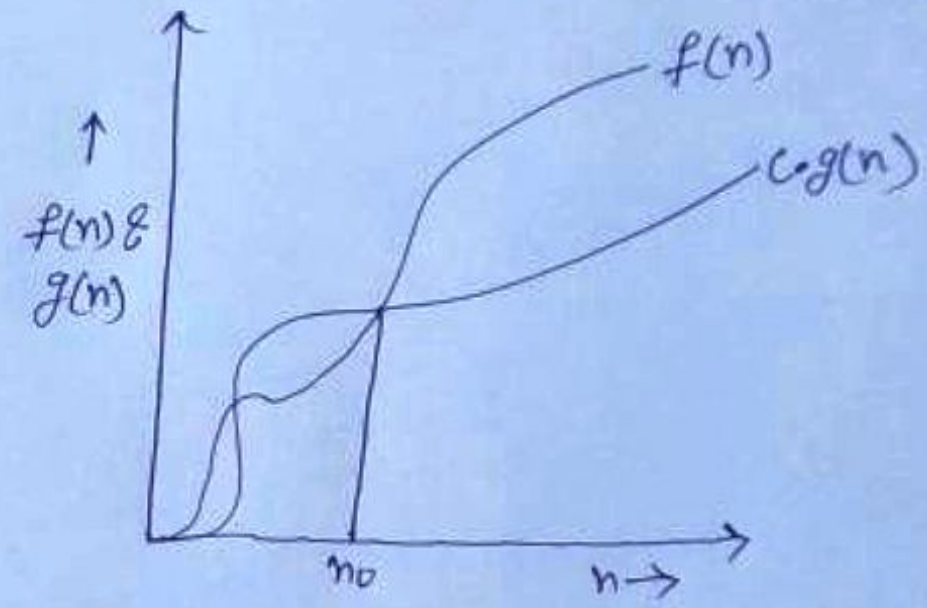
2. Big omega Notation (Ω)

Let $f(n)$ & $g(n)$ are two +ve functions.

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

iff

$f(n) \geq c \cdot g(n)$, $\forall n$, $n \geq n_0$, where c is a constant and value of $c > 0$, n_0 is constant and value of $n_0 \geq 1$.



P-1 $f(n) = n$, $g(n) = n+10$

P-2 $f(n) = n^2+n+10$, $g(n) = n^2$

P-3 $f(n) = n$, $g(n) = n^2$

3. Theta Notation (Θ)

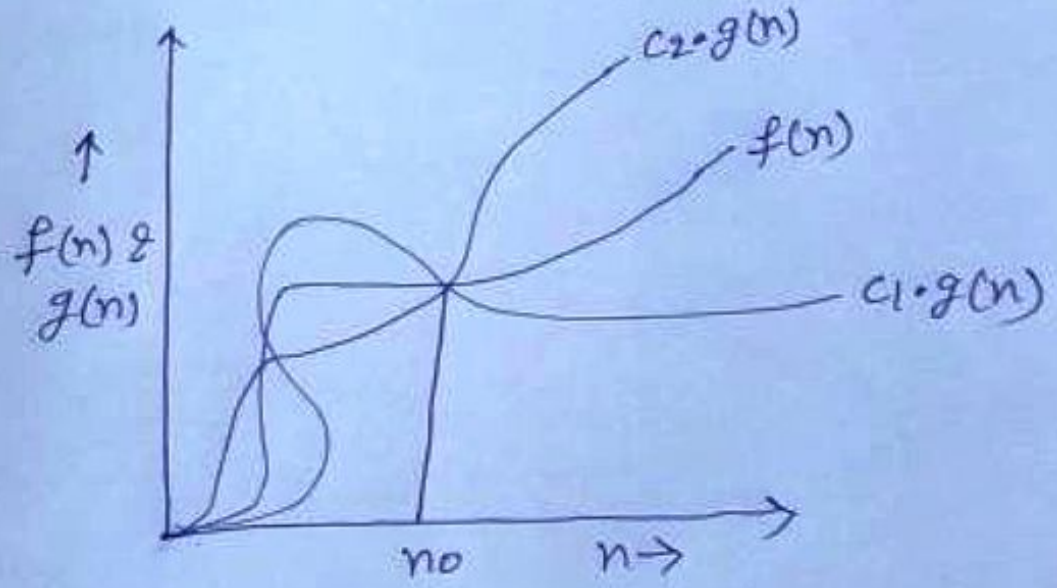
Let $f(n)$ & $g(n)$ are two +ve functions.

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

iff

$$c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n), \forall n, n \geq n_0,$$

where c_1, c_2 are constants and value of $c_1, c_2 > 0$, n_0 is a constant and value of $n_0 \geq 1$.



P-1 $f(n) = n$, $g(n) = n+10$

P-2 $f(n) = n$, $g(n) = n$

P-3 $f(n) = n^2$, $g(n) = n^2 + n + 10$

P-4 $f(n) = n^2$, $g(n) = n$