Asymptotic Notation — Time Complexity represent in form of mathematical function.

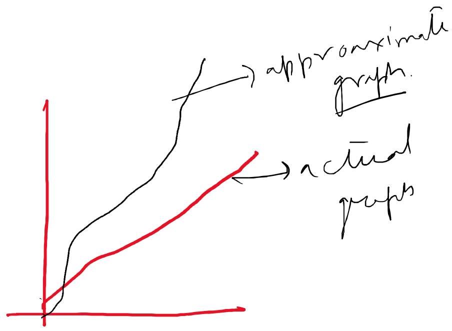
$$1 < \lg n < \sqrt{n} < n < n \lg n < n^2 < n^3 < 2^n < 3^n < n^n$$

This is true for large n.

1. Worst Case — Big OH
2. Best Case — Big Omega
3. Average Case — Theta

Big OH — Given two functions **f(n)** and **g(n)**, we say that **f(n)** is **O(g(n))** if there exist constants **c > 0** and n_0 >= 0 such that **f(n) <= c*g(n)** for all **n >=** n_0.

eg- $TC = f(n) = 2n^2 + 4n + 5 \approx c \times n^2$



TC → $f(n) = 2n+3$. Upper bound.

Ans — $f(n) = O(g(n))$

$\Rightarrow f(n) \le c \cdot g(n)$ ← $g(n) = ?$ $c = ?$ $n_0 = ?$

$g(n) = n$
$c = 5$
$n_0 = 1$

$2n+3 \le c \cdot g(n)$ ← From Defn.

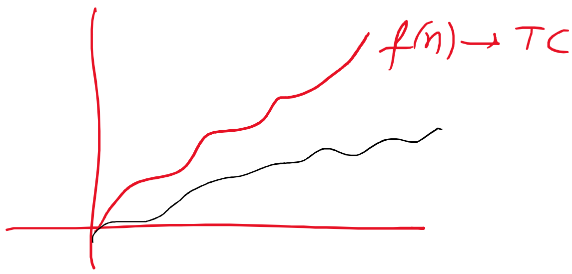
$\Rightarrow 2n+3 \le c \cdot n$
$\Rightarrow 2n+3 \le 5n$
$\Rightarrow 3 \le 3n$
$\Rightarrow n \ge 1$ $n \ge n_0$

$f(n) = 2n + n^2 \le c \cdot g(n)$
$2n + n^2 \le 3 \cdot n^2$
$2n \le 2n^2$
$\Rightarrow n \le n^2$ $\Rightarrow n^2 \ge n$
$\Rightarrow n \ge 1$

① Big Ω → Best Case

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

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



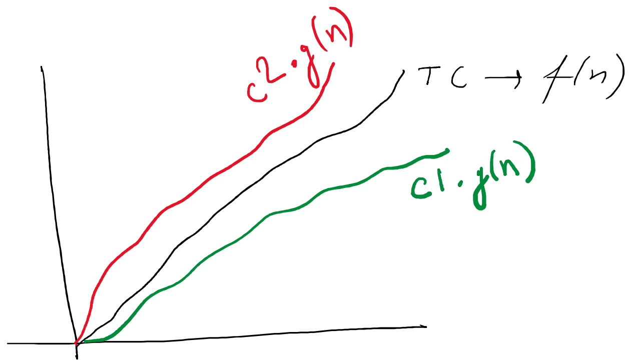
$f(n) = \Omega(g(n))$

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

③ Theta notation -

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

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



Eg - Find ~~lower bound~~ of $f(n) = 100n + 5.$

$c > 0,\ n_0 > 0$

$n \geq n_0$

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

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

$\Rightarrow c \cdot g(n) \leq 100n + 5.$

$\Rightarrow c \cdot n \leq 100n + 5.$

$105n - 100n \leq 5$

$5n <= 5$

$n <= 1$

c	n_0
105	1
210	2
10	