2 Courses

DSA ✓
- → GATE CS, DA
- → Interview prep.

Problem solving in DSA ✓
- → mainly for interview prep
- → It also helps in Gate prep

only Gate
→ DSA is enough

For Gate & Inter
→ DSA
→ PS. in DSA.

What is an algorithm:-

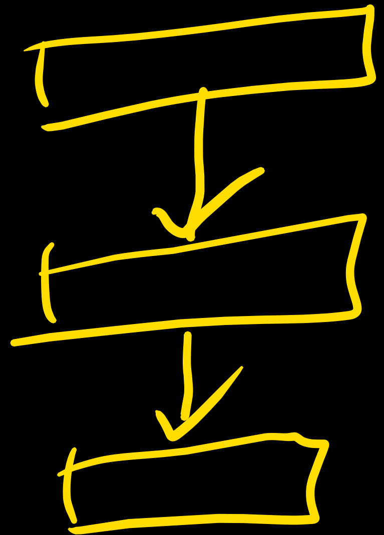
An algorithm is a step by step procedure or formula for solving a problem or accomplishing a task.

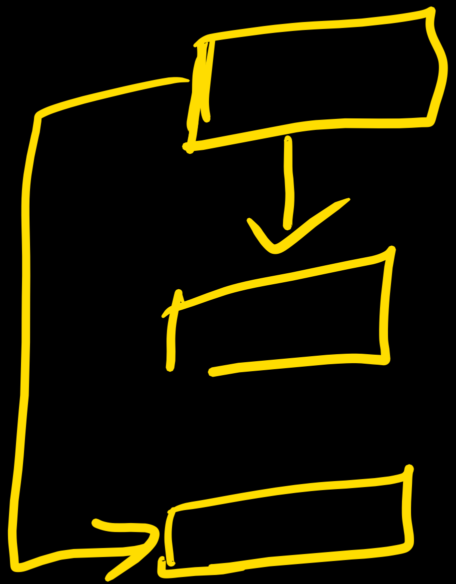
Its a sequence of well defined instructions or rules that specify how to solve a problem in finite no. of steps.

Key characteristics of algorithm:

→ **Finite:** It should terminate after a finite number of steps

→ **unambiguous:** Every step should clearly specify next step.





unambiguous.

3) <u>I/p o/p</u>: Algo takes i/p and produce o/p.

4) <u>Effective</u>: Algo should solve the problem correctly for all possible i/p's within a reasonable amount of ~~t~~ time

Problem

A_1 → $T(n)$, $S_1(n)$

A_2 → $T_2(n)$, $S_2(n)$

A_3 → $T_3(n)$, $S_3(n)$

Compare the time & space asymptotically.

Asymptotic analysis:-

$n \to \infty$ how does time and space complexity of algo behave

Formal def: It is a method used in Computer Science and mathematic to describe the limiting behaviour of a function of algorithm as its i/p size tends towards infinity.

It helps us understand how the performance of an algorithm scales with increasing i/p size.

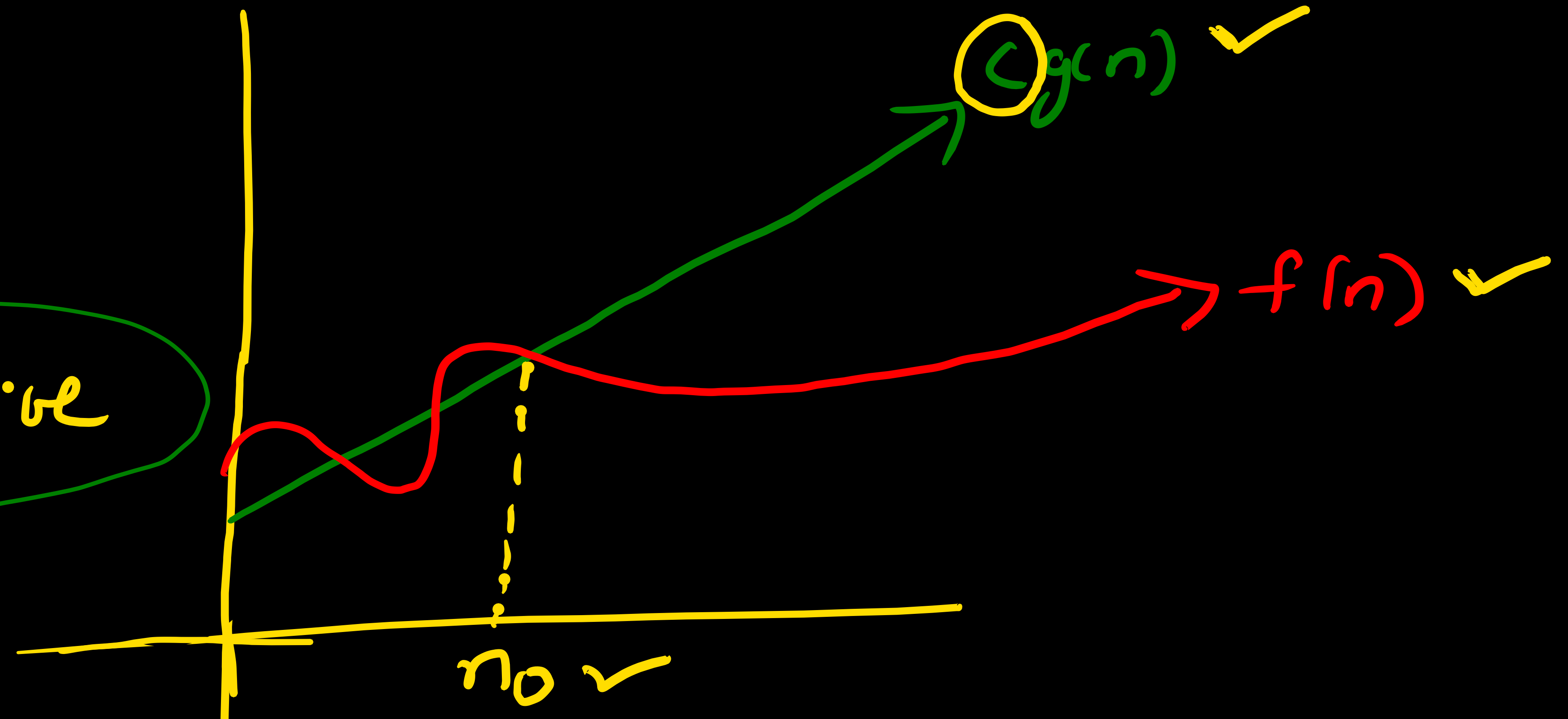
Big oh notation :- O

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

if there exists positive constant c and n_0 such that

$$f(n) \le c\,g(n) \quad \forall\, n > n_0$$

one ~~num~~ constant is sufficient



$$c,\ n_0 > 0$$

Constant   Constant

$f(n) = n^2 \qquad g(n) = n^3$

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

$n^2 = O(n^3)$

$n^2 \leq \underset{\downarrow}{c}\, n^3 \qquad n_0 = 1$

1

$f(n) = 2n^2 + 3 \qquad g(n) = n^2$

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

$f(n) \leq c\, g(n), \; n > n_0$

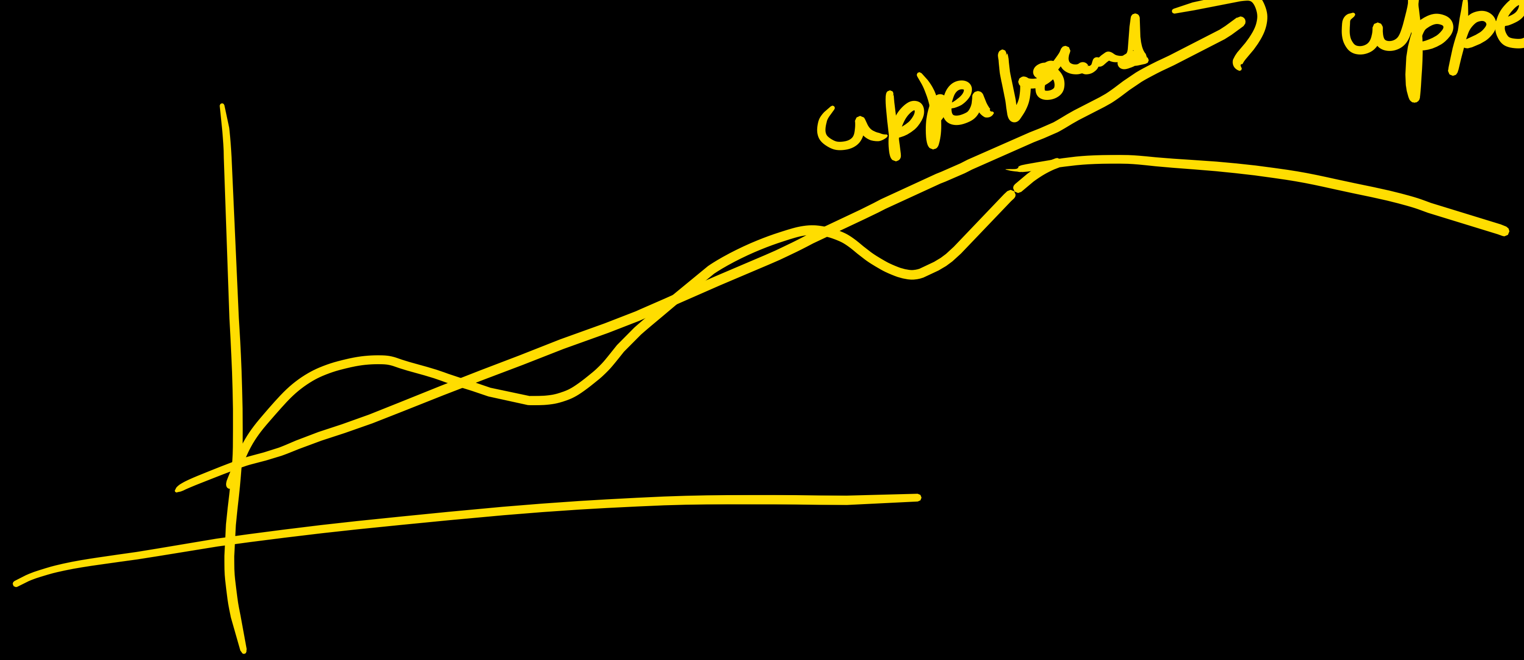
$2n^2 + 3 \leq \underset{\downarrow\, c}{\underline{10000}}\, n^2, \; \underline{n > 10000}$

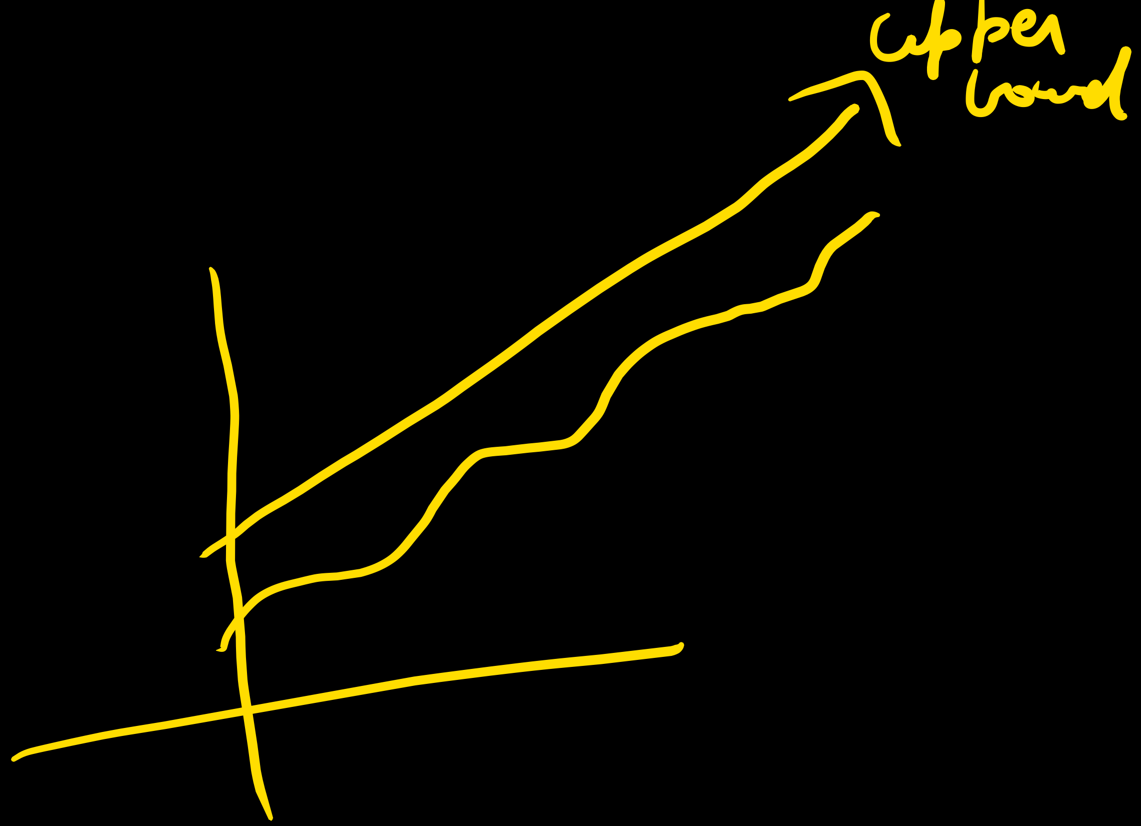
$f(n) = O(g(n))$

$f(n) = n^2 \qquad g(n) = n^2 \log n.$

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

upper bound





when we have an asymptotic upper bound we can used

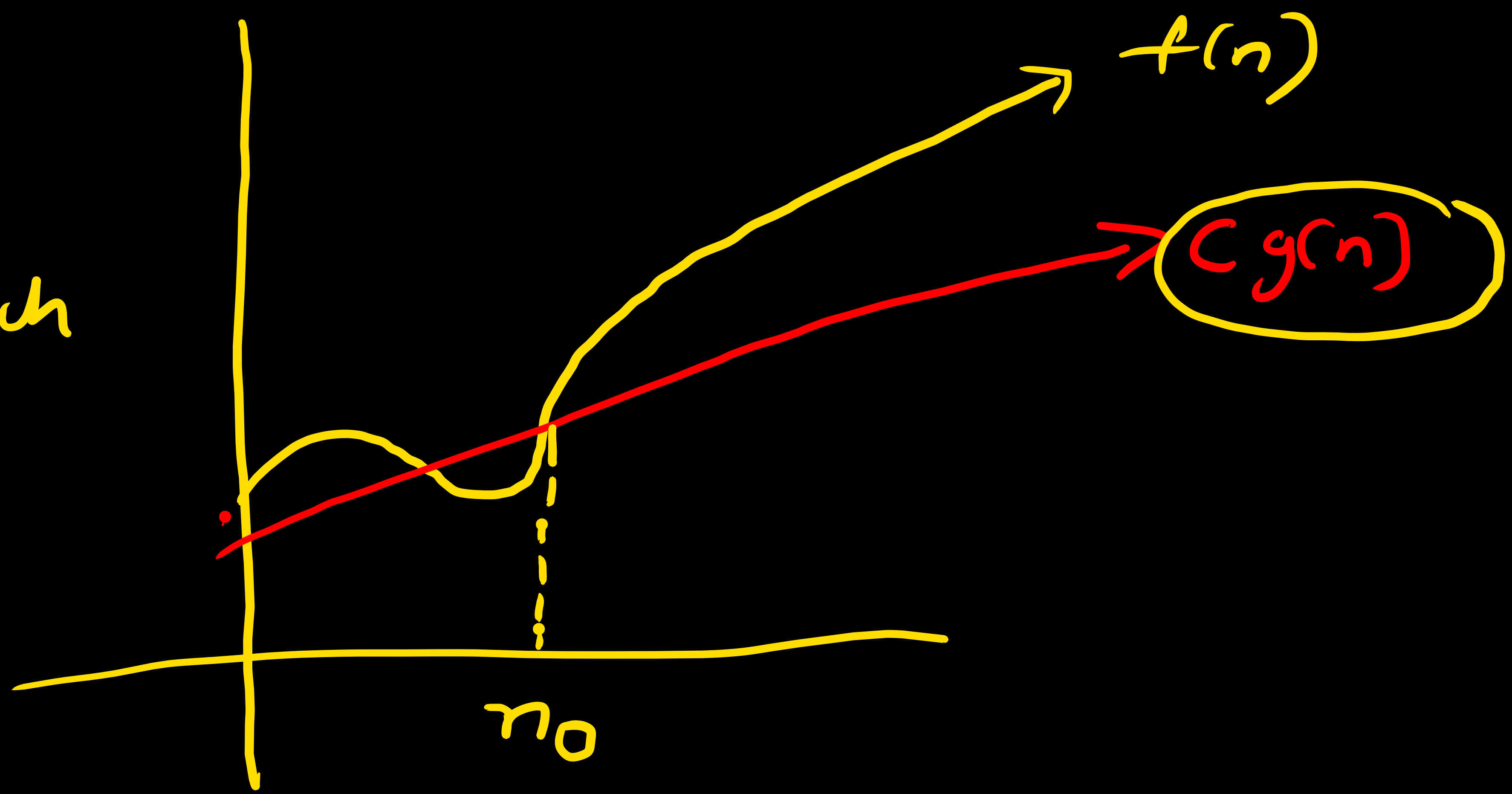
O - notation.

Big omega : Ω :

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

if there exists positive constants c and n_0 such that $f(n) \geq c\,g(n)$

for all $n \geq n_0$



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

$$f(n) = n^3 \qquad g(n) = n^2$$

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

$$n^3 \geq c\, n^2$$

$$n_0 = 1$$

$$n^3 = \Omega(n^2)$$

$$f(n) = 3n^3 + 2 \qquad g(n) = n^3$$

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

$$3n^3 + 2 \geq c\, n^3$$

$$1 \qquad n_0 = 1$$

$$f(n) = 3n^2 + 2$$

$$g(n) = n^2$$

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

$$3n^2 + 2 \leq cn^2$$

$$\downarrow$$

10000

$n_0 = 10000$

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

$$\cancel{f(n) \geq}$$

$$3n^2 + 2 \geq cn^2$$

1

$3n^2 \mid n^2$

$3n^2+2n+1 \mid n^2$

$n \mid n\log n$ only 'O'

functions differ only by constants in the leading term; then we can adjust the value c, n_0 and make them O, Ω and Θ → later

$f(n)$ & $g(n)$ → Time and Space.

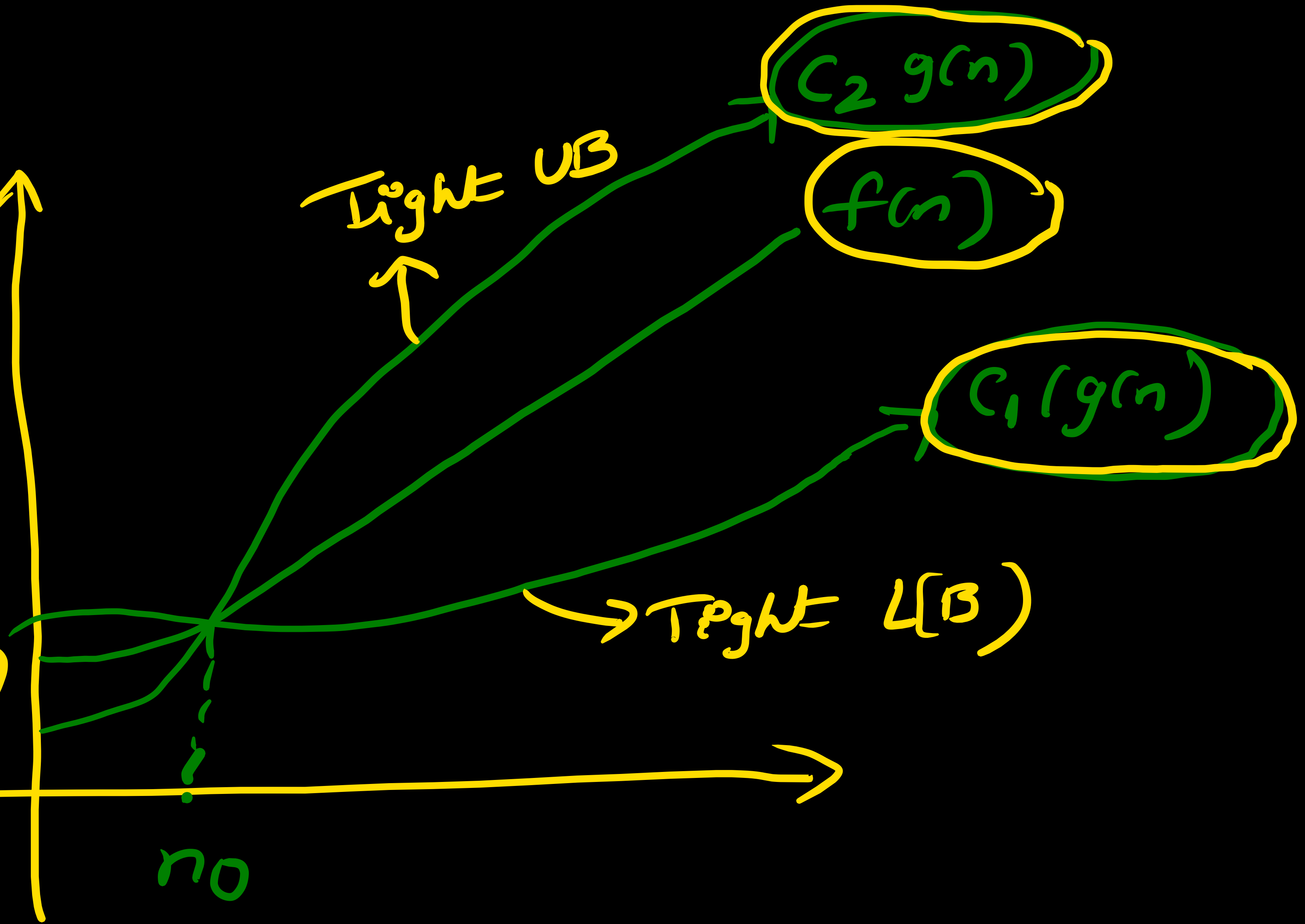
Theta (Θ):

if $f(n) = \Theta(g(n))$ then $g(n)$ is asymptotically tight bound

fn $f(n)$

$f(n) = \Theta(g(n))$ if there exists positive constants c_1 c_2 and n_0 such that

$c_1 g(n) \le f(n) \le c_2 g(n)$



$$2n^2 \le 3n^2 \le 4n^2$$

$$3n^2 = \Theta(n^2)$$

if $f(n) = O(g(n))$ ~~&&~~ $f(n) = \Omega(g(n))$

upper

lower.

$f(n) = 3n^2$

$g(n) = n^2$

$$\underbrace{2n^2}_{c_1} \leq 3n^2 \leq \underbrace{4n^2}_{c_2}$$

$n_0 = 1$

when a function is both UB and LB, it means that it is tight bound

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

Small-oh : o :

The asymptotic upper bound O (big oh) notation may or may not be asymptotically tight.

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

$f(n) = c\,g(n)$

$f(n) < c(g(n))$.

$f(n) = n^2 \quad g(n) = n^3$

$f(n) = O(g(n)) \rightarrow$ not tight

$f(n) = 3n^2 \quad g(n) = n^2$

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

tight

We use small oh (o) notation to denote an upper bound that is not asymptotically tight

<u>Ex</u>: $f(n) = n \qquad g(n) = n \log n$

$$f(n) = \underline{\underline{o}}\, g(n)$$

$$\Rightarrow f(n) < g(n)$$

$O \ \leq \qquad o \ <$

formal definition of o(n):

$f(n) = o\,g(n)$ if for any positive constant c, there exists n_0 such that $f(n) < c\,g(n)$ — for all constant

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

$f(n) \le c\,g(n)$

one value is sufficient

some n_0

$f(n) = o\,g(n)$

$f(n) < c\,g(n)$

for all values of (c, n_0)

If the difference is constant

$3n^3 \qquad n^3$ "O"

O, Ω, Θ

$3n^2 = O(n^3)$

$3n^3 < c\,n^3$

$c \downarrow 1$

when the diff is not constant

$3n^3 \qquad n^3 \log n.$

$f(n) \qquad g(n)$

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

$3n^3 < c\; n^3 \log n$ ✓

c: $0.0001,$

$n_0 = 1000000000000$

$n \geq n_0$

small omega: (ω):

we use ω to denote a lower bound that is not asymptotically tight.

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

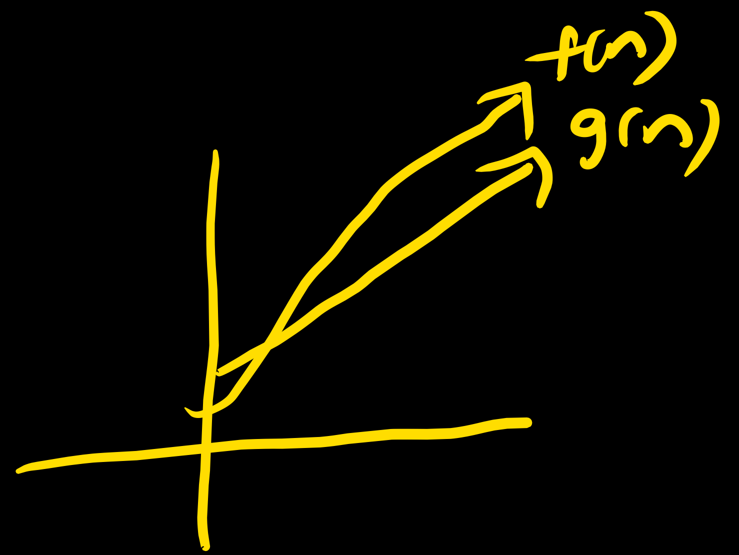
if for any positive constant $c > 0$, there exists a $n_0 > 0$

such that $f(n) > c(g(n))$

$$f(n) = \omega(g(n)) \qquad \lim_{n \to \infty} \frac{g(n)}{f(n)} = 0$$

greater.

$$\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$$



ex $g(n) = n^2 + 2$

$f(n) = n^3$

$$\lim_{n\to\infty} \frac{g(n)}{f(n)} = \frac{n^2+2}{n^3} = \frac{1}{n} + \frac{2}{n^3} = 0.$$

P

	A_1	A_2	A_3	A_4
TC :	$n!$	2^n	n^n	$(\log n)^n$

which one is ~~better~~ better

which one is upper bound

and which one is lower bound

which are asymptotically same

$$\underbrace{10000\,n^2 \qquad 2n^2}_{\text{asymptotically same}}$$

$$(10000\,n^2) = \Theta(2n^2)$$

Lot of examples to make it clear:

Complexity classes:

$DF < CF < LF.$

1. Decreasing functions

$O(1/n) \quad O(1/n^2) \quad O(1/n^3)$

2. Constant function

$$O(1) = O(1000) = O(100000)$$

3. Logarithmic functions

$$O(\log n)$$

4. linear function $O(n)$

5. quadratic function $O(n^2)$

6. cubic function $O(n^3)$

$\vdots$

7. Polinomial function $O(n^k)$

8. exponential function $O(c^n)$

Q)

log n $\sqrt{n}$

$f(n)$ $g(n)$

big omega.

small omega

apply log on both sides

$\log\log n < \cancel{1/2} \log n$

$\log n < \sqrt{n}$ (it is not constant diff

$\sqrt{n} = \omega(\log n)$ $\log n = o(\sqrt{n})$

$\sqrt{n} = \Omega(\log n)$ $\log n = O(\sqrt{n})$

Q) $n^{3/2}$ $\quad$ $n\log n$

<u>Rule</u>: Dont apply log directly, if there are some factors to be cancelled out, then do the cancellation first

$$\cancel{n}\sqrt{n} > \cancel{n}\log n$$

$$n^{3/2} > n\log n \quad \text{(not by a constant)}$$

$n^{3/2} = \omega(n\log n)$	$(n\log n) = o(n^{3/2})$
$n^{3/2} = \Omega(n\log n)$	$(n\log n) = O(n^{3/2})$

if small ~~oh~~ oh, then big oh also

But if O (big-oh), may ~~or~~ or may not be small oh)

if small omega then big omega

but if Big omega then may or may not be small omega

Q) 2^n n^n

2^n	n^n
$2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \; 2 \ldots n$ time	$n \cdot n \cdot n \cdot n \cdot n \ldots n$ times.

$2^n < n^n$ (diff is not by constant)

$n_0 \geq 3$ $2^n = o(n^n)$

$2^n = O(n^n)$

$n!$

$1 \cdot 2 \cdot 3 \cdot 4 \cdots\cdots n$

$4!$

$4 \times 3 \times 2 \times 1 = 24.$

n^n

$n \cdot n\, n \cdot n \cdots\cdots$ n times

4^4

$= 256$

$n! < n^n$

$\cancel{\ } \; n! = o(n^n)$

$n! = O(n^n)$

$\rightarrow$ 2^n

$2 \cdot 2 \cdot 2 \cdot 2 \cdots\cdots 2$

2^5

$2 \cdot 2 \cdot 2 \cdot 2 \cdot 2$

$n!$

$n(n-1)(n-2)\cdots\cdots 2 \cdot 1$

$5!$

$5 \times 4 \times 3 \times 2 \times 1$

for small values

$n! < 2^n$

For large value.

$n \geq n_0$

n_0 can be any value

$n_0 = 10000$

$= 10^{10}$

$2^n < n!$ (diff is not constant)

$2^n = o(n!)$

$\therefore$

$2^n = O(n!)$

Q)

$\log_2 n \qquad \log_3 n$

$$\log_b a = \frac{\log_c a}{\log_c b}$$

$\log_2 n \qquad \log_3 n = \dfrac{\log_2 n}{\log_2 3}$

$$= \frac{\log_2 n}{\cancel{1.5}\ 1.5}$$

$$\log_2 n = \Theta(\log_3 n)$$

$$\log_2 n = O(\log_3 n)$$

$$\log_2 n = \Omega(\log_3 n)$$

diff is constant:

$$\log_2 n = O\left(\frac{\log_2 n}{\cancel{1.5}\ 1.5}\right)$$

0.747

$$\log_2 n = o\left(\frac{\log_2 n}{1.5}\right)$$

$$\log_2 n < c\,\frac{\log_2 n}{1.5} \times$$

$o \times$ $\cancel{1.5}$!

$\Omega \times$

$$\log_2 n \le c\,\frac{\log_2 n}{1.5}.$$

1.5

one value

$$\log_2 n = \Omega\left(\frac{\log_2 n}{\cancel{1.5}\ \cancel{0.44}}\ 1.5\right)$$

$$\log_2 n \geq c\,\frac{\log_2 n}{\cancel{1.5}\ \cancel{0.47}\ 1.5}$$

$$\downarrow$$

1.5 ✓

$O \quad \Omega \quad \Theta.$

$\times\, o \quad \times\, \omega$

$$\log_{10} n = \Theta(\log_{25} n)$$

$$\log_2 3.$$

$$\log_{100} n = \Theta(\log_{200} n)$$

If O Ω, then it is Θ also.

If it Θ, then it is O and Ω also.

How O Ω ✓

$\leq$ $\geq$

$=$

if of diff is Constant, we get $=$

when we get '$=$' O Ω

Q) 2^n 3^n

$\Downarrow$ $\Downarrow$

Is it constant difference ?

2 2 2 2 2 ... n times 3 . 3 . 3 . 3 ...... n time

I need to know if the diff is constant

Dont apply log directly.

$\cancel{\log 2^n \quad \log 3^n}$

$\cancel{n \log 2 \quad n \log 3}$ → constant diff

$2^n \qquad 3^n$

$2^n \qquad (2 \times 1.5)^n$

$1 \quad \cancel{2^n} \qquad \cancel{2^n}(1.5)^n$

diff is not const.

$2^n = o(3^n)$

$2^n = O(3^n)$

$2^n \neq \omega(3^n)$

$2^n \neq \Omega(3^n)$

$2^n \neq \Theta(3^n)$

$$(\log n)^{1000000000} \quad , \quad n$$

apply log.

$$\cancel{1000000000}\ \log\log n < \log n.$$

- n is greater not by a constant.

$$(\log n)^k = o(n)$$
$$= O(n)$$

$$(\log n)^{\log n} > n$$

$$\cancel{\log n}\, \log\log n > \cancel{\log n}\, 1$$

$$n < (\log n)^{\log n}$$

$$n = o(\log n)^{\log n}$$

$$= O(\log n)^{\log n}$$

$n^2 \qquad n^3$

Dont apply log directly

$\cancel{2 \log n} \qquad \cancel{3 \log n}$

$\cancel{1} \qquad \cancel{1}$

$\cancel{n^2 = O(n^3)}$

$\cancel{n^2 = \Omega(n^3)}$

$\cancel{n^2 = \Theta(n^3)}$

$n^2 \qquad n^3$

$\cancel{n^2} \qquad \cancel{n^2}\, n$

$\underbrace{1 \qquad n}$

diff is not constant.

$n^2 = o(n^3) \qquad n^2 \neq \omega(n^3)$

$n^2 = O(n^3) \qquad n^2 \neq \Omega(n^3)$

$n^2 = O(n^3) \qquad n^2 \neq \Theta(n^3).$