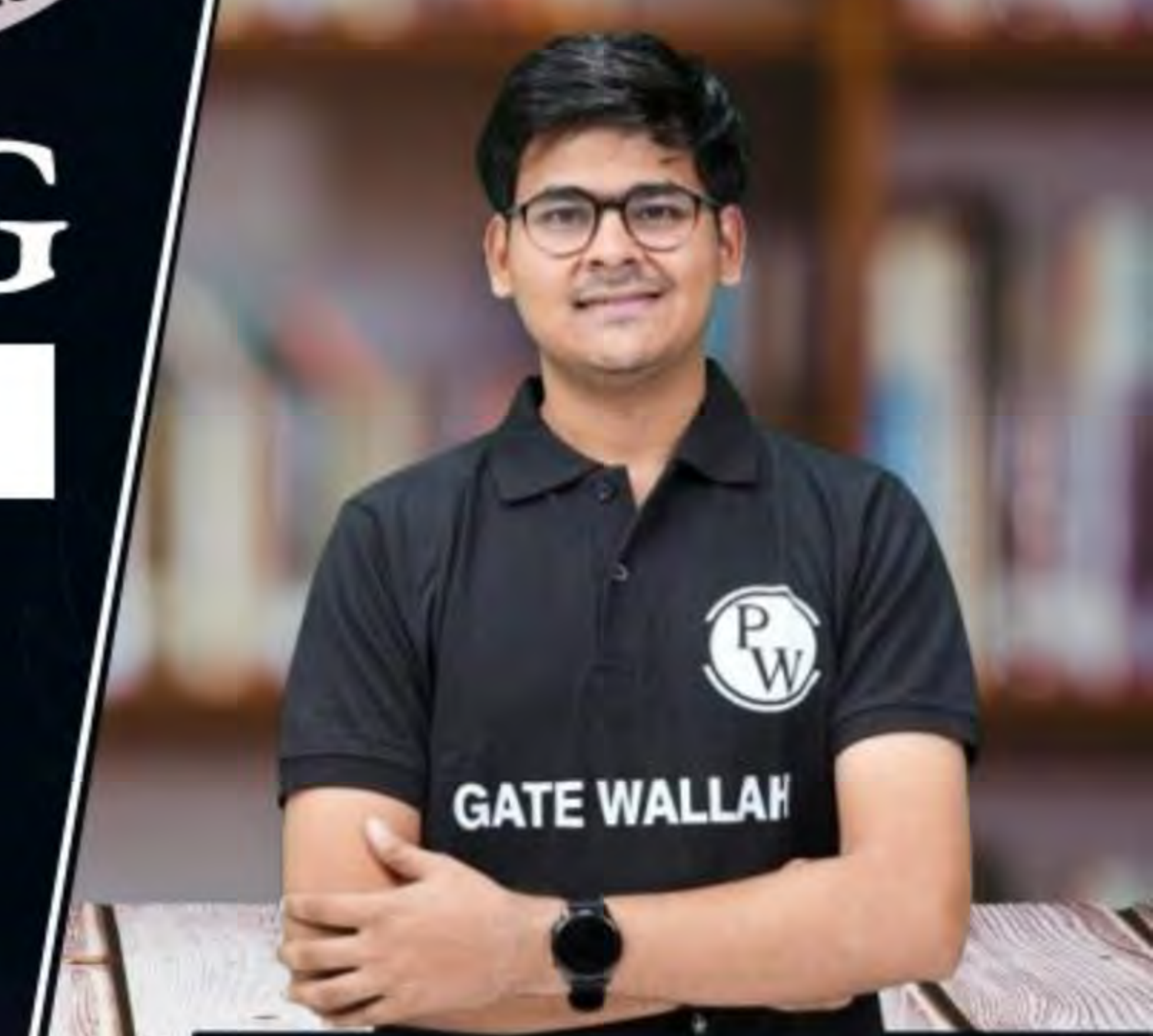
CS & IT ENGINEERING

Algorithms

Analysis of Algorithms

Lecture No.- ~~06~~ 07



By- Aditya sir

Recap of Previous Lecture

Topic

general

Tricotomy Prop

Topic — Properties of Asymptotic Notation

Topic — Problem Solving

Imp Questions

Topics to be Covered

Topic Problem Solving with ASN (PYQs + Imp Questn)

Topic Framework for Analysing Non-Recursive algorithm

Discuss Properties of Asymptotic Notations

* Discrete Properties of Asymptotic Notations.

if $f = O(g)$ then $g = \Omega(f)$

$f < g$ then $g > f$

if $f(n) = o(g)$ then $g = \omega(f)$

	Reflexive	Symmetric	Transitive	☆ Transpose Symmetry
O	✓	✗	✓	✓
Ω	✓	✗	✓	✓
Θ	✓	✓	✓	✗
o	✗	✗	✓	✓
ω	✗	✗	✓	✓

Symmetry

If $n = O(n^2)$

then $n^2 = O(n) \rightarrow$?

X

if $f = \Theta(g)$

then $g = \Theta(f)$ ✓

Transitive

$a \le b$
& $b \le c$ $\Big\} \Rightarrow a \le c$

$a < b$
& $b < c$ $\Big\} \Rightarrow a < b < c$

✓

Transpose Symmetry :-

Real nos

a, b

$a \geq b \Rightarrow b \leq a$?

$\longleftrightarrow$

Asymptotic Notations.

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

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

Let a, b be two real numbers

& f, g be two +ve functions of n.

$f \longrightarrow a$

$g \longrightarrow b$

Asymptotic Notations $\Longrightarrow$ Real numbers.

① if $f(n) = O(g(n)) \Longrightarrow a \leq b$

② if $f(n) = \Omega(g(n)) \Longrightarrow a \geq b$

③ if $f(n) = \Theta(g(n)) \Longrightarrow a = b$

④ if $f(n) = o(g(n)) \Longrightarrow a < b$

⑤ if $f(n) = \omega(g(n)) \Longrightarrow a > b$

Topic : Asymptotic Comparisons

V. Imp

f(n), g(n) : are functions

$\boxed{f(n) = O(g(n))}$ → Given

T/F a) Is $f(n) = O(f(n)^2)$ → False

T/F b) $2^{f(n)} = O(2^{g(n)})$ → False

Check

A) only a is True → 13.3% ✗

B) only b is True → 26% ✗

C) both a & b are True → 56% ✗

D) both a & b are False → 5%

✗ speed > Accuracy

✓ Accuracy > speed

given: $f(n) = O(g(n)) \longrightarrow f \le c * g$

eg1:- $f(n) = n$

Check a) $(f(n))^2 = n^2$

Is $f(n) = O(f(n)^2)$?

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

true

eg2:- $f(n) = 1/n$

$(f(n))^2 = 1/n^2$

Is $f(n) = O(f(n)^2)$?

Is $1/n = O(1/n^2)$?

$\longrightarrow$ No.

Check b):

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

Check if $2^{f(n)} = O(2^{g(n)})$?

eg1:- $f(n) = n$

$g(n) = n^2$ ⟶ Is $2^{f(n)} = O(2^{g(n)})$?

valid eg ?

Is $2^n = O(2^{n^2})$?

Yes ⟶ True

$2^n < 2^{n^2}$

$f(n) = n$

$g(n) = 10$?

$\}$ ⟶ invalid eg.

eg2:

given
$f = O(g)$ ✓

$F(n) = 2n$
$g(n) = n$

$2^{f(n)} = O(2^{g(n)})$?

False

Fails for this Case

$2^{(2n)} = O(2^n)$?

2^{2n} 2^n

way1

$2^n \cdot \cancel{2^n}$ $\cancel{2^n}$

$2^n > 1$

OR

way2

$\log_2()$

$2n$ n

$n + \cancel{n}$ $\cancel{n}$

$n > 1$

[MCQ]

Gate 2022

#Q. Which one of the following statements is True for all positive functions $f(n)$?

44%

A $f(n^2) = \theta(f(n)^2)$, when $f(n)$ is a polynomial ✓

Small oh

B $f(n^2) = o(f(n)^2)$ $\longrightarrow$ ✗

Big Oh

C $f(n^2) = O(f(n)^2)$, when $f(n)$ is an exponential function $\rightarrow$ ✗

D $f(n^2) = \Omega(f(n^2))$ $\longrightarrow$ False

Ans: A

$f(n) \longrightarrow$ Polynomial

Check A :-

$$f(n^2) = \Theta\left(f(n)^2\right) \checkmark$$

$$(a^m)^n = (a^n)^m$$
$$= a^{(m \times n)}$$

eg:- $f(n) = n^3$

$f(n^2) = (n^2)^3 = \underline{\underline{n^6}}$

$\Rightarrow (f(n))^2 = (n^3)^2$
$= n^{3 \times 2}$
$= \underline{\underline{n^6}}$

B) $f(n) = n$ $\quad (f(n))^2 = n^2$

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

Is $f(n^2) = O((f(n))^2)$?

$n^2 = O(n^2)$? $\longrightarrow$ YES

c) Is $f(n^2) = O(f(n)^2)$ when $f(n)$ is exponential?

eg:- $f(n) = 2^n$

$f(n^2) = 2^{(n^2)}$

$(f(n))^2 = (2^n)^2 = 2^{2\times n} = 2^{2n}$

2^{n^2} > vs 2^{2n}

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

$n > 2$

$f(n^2) \neq O((f(n))^2)$

in this case

Is $F(n^2) = \Omega((F(n))^2)$?

Check D :

$f(n) = n^3$

$f(n^2) = (n^2)^3 = n^6$

$(f(n))^2 = (n^3)^2 = n^6$

$F(n^2) = (F(n))^2$

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

true

Case 2 : $f(n) = \log(n)$

$f(n^2) : \log(n^2)$

$(f(n))^2 = (\log n)^2$

$\log(n^2) \qquad (\log n)^2$

$2 * \cancel{\log(n)} \qquad (\log n) * \cancel{(\log n)}$

$2 < \log n$

In this case

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

Hence $f(n) = \Omega(F(n)^2)$ fails (false)

[MCQ] $\longrightarrow$ MSQ

PYQ

$f < g$

$\hookrightarrow f \le g$

#Q. Which of the following is TRUE.

1. f(n) is O(g(n))
2. g(n) is NOT O(f(n))
3. g(n) is O(h(n))
4. h(n) is O(g(n))

given

$f + h = O(g + h)$

$f + \cancel{h} \le g + \cancel{h}$

$f \le g$

A f(n) is O(h(n)) $\rightarrow$ True

B h(n) $\neq$ O(f(n)) $\rightarrow$ True

C f(n) + h(n) is O(g(n) + h(n))

True

D f(n).g(n) $\neq$ O(g(n). h(n)) $\rightarrow$ False

$2 < 3$ $\quad$ $2 \le 3 \rightarrow$ less or equal given $\longrightarrow$ Conclusion

Sdn :- 1) $f(n) = O(g(n)) \Longrightarrow F(n) \le g(n)$

2) $g(n) \ne O(f(n)) \Longrightarrow g(n) \not\le f(n) \Rightarrow g(n) > F(n)$

$\Downarrow$

3) $g(n) = O(h(n)) \Longrightarrow g(n) \le h(n)$

4) $h(n) = O(g(n)) \Longrightarrow h(n) \le g(n) \Longrightarrow$

$$\left.\begin{array}{l} g(n) > f(n) \quad c_1 \\ g(n) = h(n) \quad c_2 \end{array}\right\} f < (g = h)$$

Final Conclusion

Check A) $F(n) = O(h(n))$?

$F \le h(n)$ ✓

Check B

$h(n) \ne O(f(n))$

$h(n) \not\le f(n)$

$h(n) > F(n)$? ✓

$f < (g = h) \longrightarrow f < h$

Sdn : check D

$f(n) * g(n) \neq O(g(n) * h(n))$

$f \times g \cancel{\leq} g \times h$

$f \times \cancel{g} > \cancel{g} \times h \implies f > h \rightarrow$ False

[MCQ]

#Q. $f(n) = 2^n$; $g(n) = n^n$ $\Longrightarrow$ $\boxed{f < g}$

$n \geq n_0$

large values of n

65%

$f \leq g$

Ans = A

$f \geq g$ ✗

n_0

→ 65%

A) $f(n) = O(g(n))$ → True

C) $f(n) = \Omega(g(n))$ → 12%

B) $f(n) = \theta(g(n))$ ✗ → 11.2%

D) ~~None of these~~ → 11%

$f = g$

Sdn :

$f(n) = 2^n$

$g(n) = n^n$

$2^n < n^n$

2^n vs n^n

$\log_2(2^n)$ $\log_2(n^n)$

$\cancel{n} \times 1$ $\cancel{n} \times \log n$

$1 < \log n$

V. Imp

① * Every Small oh is also Big oh But
Every big oh may or may not be small oh.

② * Every Small Omega is also Big Omega
But Every big Omega may or may not be small omega.

given $a < b \Rightarrow a \le b$ ✓
But $a \le b \rightarrow a < b$?
Not necessary.
eg: $2 \le 2 \rightarrow 2 < 2$ ✗

[~~MCQ~~] MSQ → multiple can be correct. :)

#Q. $f(n) = n.2^n$; $g(n) = 4^n$

$\implies \boxed{f < g} \to f = O(g)$ ✓ $> 80\%$

62%

$f \leq g$ ✓

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

B. $f(n) = \theta(g(n))$

$f = g$ ✗

$f \geq g$ ✗

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

✗ D. None of these

Soln :-

$f(n) = n * 2^n$

$g(n) = 4^n$

$n * 2^n$ vs 4^n

Take $\log_2()$ both sides

$\boxed{\log(ab) = \log(a) * \log(b)}$

$\log_2(n \times 2^n)$ $\quad$ $\log_2(4^n)$

$\log_2(n) + \log_2(2^n)$ $\quad$ $n \times \log_2(4)$

$\log_2(n) + n$ $\quad$ $2n$

$\log n + \cancel{n}$ $\quad$ $\cancel{n} + n$

$\log n < n$

$f < g$

[MCQ]

#Q. Let w(n) and A(n) represent respectively, the worst case and average case running time of an algorithm with input size of n, Which is always TRUE?

$$\boxed{B(n) \le A(n) \le \omega(n)} \longrightarrow \text{general}$$

Ans:- only D

A. $A(n) = o(w(n))$ — $A < W$ (sometimes True)

B. $A(n) = \theta(w(n)) \rightarrow A = W$ (sometimes True)

C. $A(n) = \Omega(w(n))$ — $A \geq W \longrightarrow$ Sometimes True

D. $A(n) = O(w(n))$ — Big oh (as can be equal at times) — $A \leq W \longrightarrow$ Always True

E. $A(n) = \omega(w(n))$ — $A > W \longrightarrow$ Never True

2 mins Summary

Topic Problem Solving with ASN ✓

Topic Framework for Analysing Non- Recursive algorithm → Next lec

THANK - YOU

Telegram Link: https://t.me/AdityaSir_PW