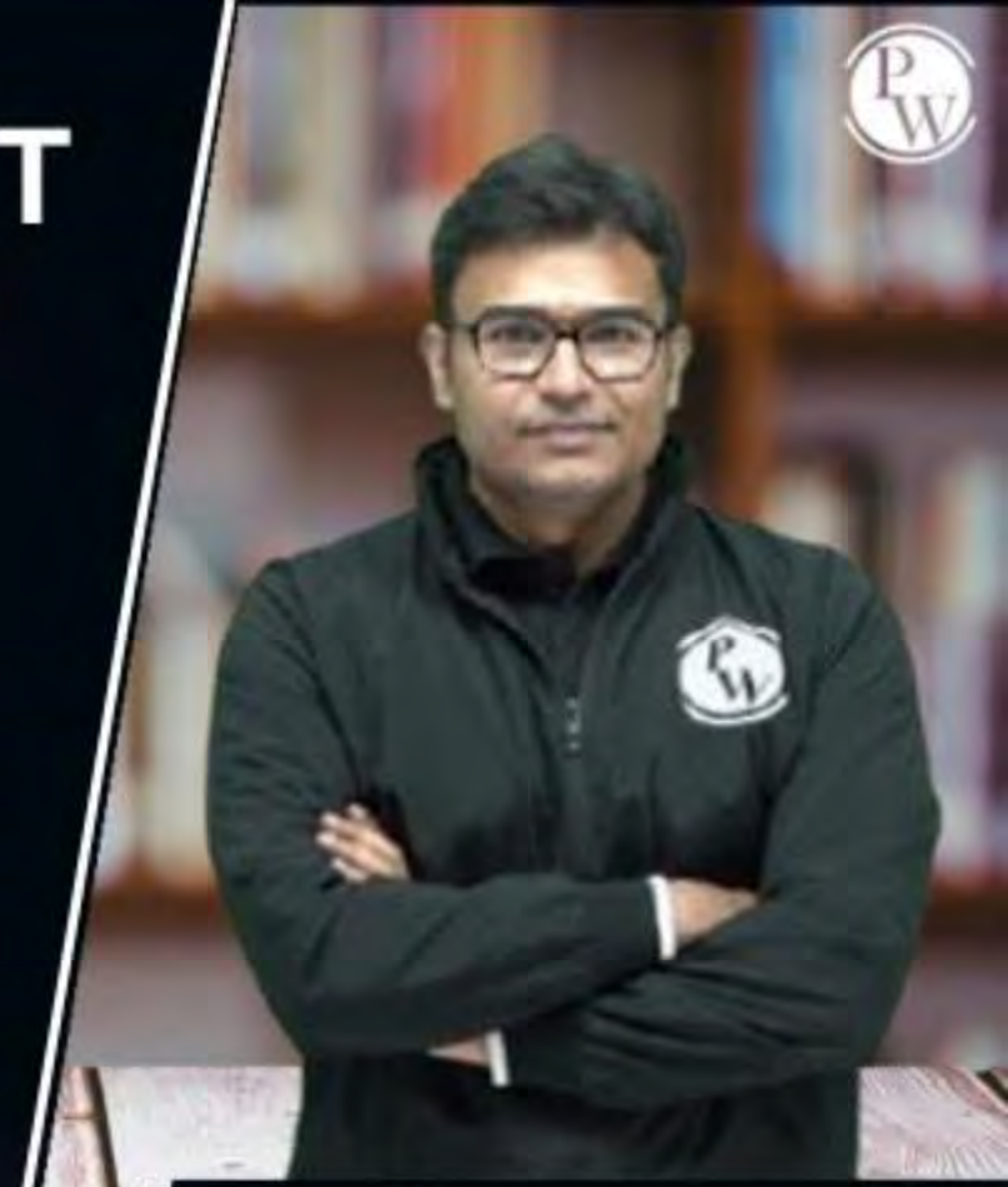
Computer Science & IT

ALGORITHMS



Algorithm

Lecture No. 4

By- Ravindra Sir

Recap of Previous Lecture

Topic TC

Topic

Topic

Topics to be Covered

Topic — Substitution method

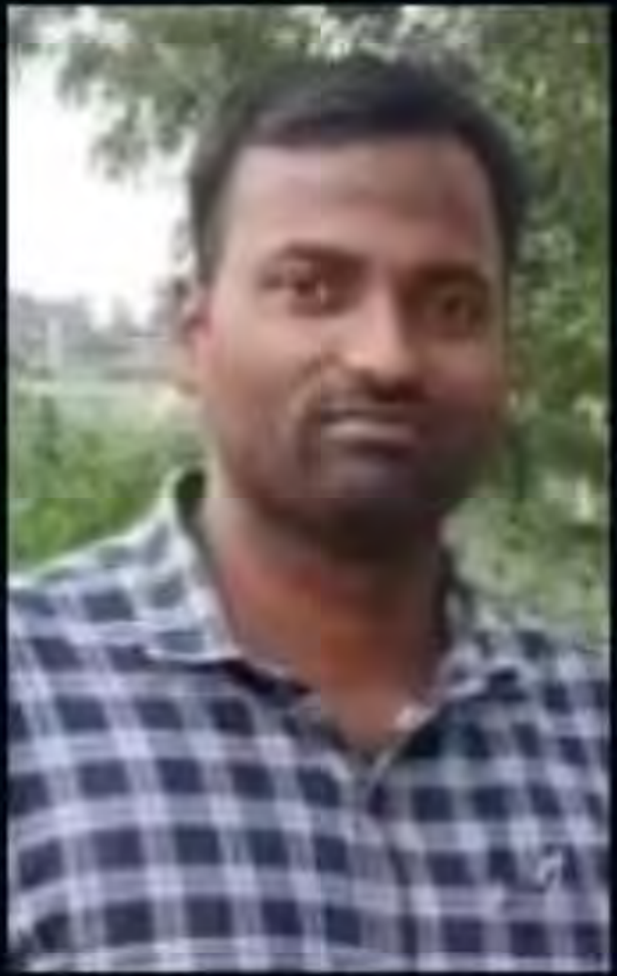
Topic

Topic

Join

@RBRSIRPW

Inspiring Stories : Raju Mupparapu



Background: From Warangal, Telangana. Noticed lights were on even when nobody was around.

Education: Local innovator, no formal education.

Achievements: Built a streetlight system using light sensors (LDR) to turn off lamps when not needed.

Impact: Cut electricity costs by 30% in many villages—saving money and power.

Inspiring Stories : Bommai N

Background: In Karnataka, saw his mother struggle to make tons of rotis daily.

Education: Grew up in a cycle shop, but curious and creative.

Achievements: Invented a handheld roti maker under 6 kg that flattens 180 rotis per hour using a simple lever system.

Impact: Eases the toughest kitchen task for rural women, saving time, energy, and effort.

Inspiring Stories : Moa Subong

Background: Musician from Nagaland. Wanted an easy-to-play instrument that sounded like home.

Education: Artisan, self-trained in music.

Achievements: Built Bumhum, a bamboo instrument that anyone can pick up and play, even without training.

Impact: Preserves indigenous Naga music and culture, making it accessible to young and old alike.

let $n/2^k = 1 \Rightarrow k = \log n$

$$n/2^{k-1} = n/2^k * 2 = 2$$

$$T(n) = 1 + 2^1 + 2^2 + \cdots\cdots + 2^K$$

$$= 1 + 2^1 + 2^2 + \cdots - - + 2^{\log n}$$

$$= 1 \frac{\left(2^{\log_2 n + 1} - 1\right)}{2 - 1}$$

$$a^{\log_b c} = c^{\log_b a}$$

$$= O\left(2^{\log_2 n}\right) = n^{\log_2 2} = O(n)$$

$$T(n) = \begin{cases} 1 & ; n = 1 \\ 2T(n/2) + n & ; n > 1 \end{cases}$$

$$T(n) = 2T(n/2) + n \rightarrow (1)$$

$$T(n/2) = 2T(n/4) + n/2 \rightarrow (2)$$

$$T(n/4) = 2T(n/8) + n/4 \rightarrow (3)$$

(2) in (1)

$$T(n) = 2\left(2T(n/4) + n/2\right) + n$$

$$= 2^2 T(n/2^2) + n + n \quad 2n \rightarrow (4)$$

$\}$ k tim

$$= 8^k T(n/2^k) + 2^{k-1} n^2 + 2^{k-2} n^2 + \cdots + 2^2 n^2 + 2 n^2 + n^2$$

$$= 8^k T(n/2^k) + n^2 (2^{k-1} + 2^{k-2} + \cdots\cdots 2^2 + 2^1 + 2^0)$$

let $n/2^k = 1 \Rightarrow \underline{k} = \log n$ $\quad SC = (\log n)$

$$= 8^{\log_2 n} + n^2 (2^0 + 2^1 + 2^2 + \cdots 2^{\log n - 1})$$

$$= n^{\log_2 8} + n^2 \left(\frac{1(2^{\log_2 n} - 1)}{2 - 1} \right)$$

$$= n^3 + n^2 (n-1) = O(n^3) = \Omega(n^3) = \Theta(n^3)$$

$$\frac{n}{2^k} = 2 \Rightarrow k = \log n - 1$$

$$= 7^{\log n - 1}(2) + n^2\left(\left(\tfrac{7}{4}\right)^0 + \left(\tfrac{7}{4}\right)^1 + \cdots \left(\tfrac{7}{4}\right)^{k-1}\right)$$

$$= 7^{\log n - 1}(2) + n^2\left(\left(\tfrac{7}{4}\right)^0 + \left(\tfrac{7}{4}\right)^1 + \cdots \left(\tfrac{7}{4}\right)^{k-2}\right)$$

$$= \frac{7^{\log n}}{7} \cdot 2 + n^2\left(\frac{\left(\tfrac{7}{4}\right)^{k-1} - 1}{\left(\tfrac{7}{4}\right) - 1}\right)$$

$$= n^{\log_2 7} + n^2 \left(\tfrac{7}{4}\right)^{\log n - 1}$$

$$O\left(n^{\log_2 7}\right) = O\left(n^{2.81}\right)$$

$= 2^3 T(n/2^3) + 3n$

$\Big\downarrow$ k times

$= 2^k T(n/2^k) + kn$

let $n/2^k = 1 \Rightarrow k = \log_2 n$

$n(1) + \log_2 n \; n$

$\underline{O(n\log n)}$ $\underline{\theta(n\log n)}$ $\underline{\Omega(n\log n)}$

$\}$ k times

$$= 2^k T\left(\frac{n}{2^k}\right) + n\log\frac{n}{2^{k-1}} + n\log\frac{n}{2^{k-2}} + \cdots + n\log n$$

let $\frac{n}{2^k} = 1 \Rightarrow k = \log_2 n$

$$= n + n\left(\log\frac{n}{2^0} + \log\frac{n}{2^1} + \log\frac{n}{2^2} + \cdots + \log\frac{n}{2^{k-1}}\right)$$

$$= n + n\left(\log n - \log 2^0 + \log n - \log 2^1 + \log n - \log 2^2 + \cdots + \log n - \log 2^{k-1}\right)$$

$n\log n$

$$= n + n(\log\log n) - (\;)$$

$$O(n(\log n)^2 = \Omega(n(\log n)^2) = \Theta(n(\log n)^2)$$

$$T(n) = \begin{cases} 1 & \text{if } n=1 \\ 2T(n-1)+n & \text{if } n>1 \end{cases} \times$$

$\underline{O(2^n)}$ $\underline{\Omega(2^n)}$ $\underline{\Theta(2^n)}$

Very difficult

Combination of <u>AP and GP</u>

Telescopic method ×

$$= \frac{n}{2} T(2) + \underline{k}\, n$$

$$n^{1/2^k} = 2$$

$$\frac{1}{2^k} \log n = \log 2$$

$$2^k = \log n$$

$$\underline{k} = \underline{\log \log n} \qquad SC = O(\log \log n$$

$$= n + n \log \log n$$

$$O(n \log \log n) \quad \Omega(n \log \log n) \quad \Theta(n \log \log n)$$

$$T(n) = 2\underbrace{T(n-1)}_{\text{only one term}} + n$$

only one term → Substitution ✓

$$T(n) = T\underbrace{\left(\tfrac{n}{3}\right) + T\left(\tfrac{2n}{3}\right)}_{\text{more than one term}} + n$$

more than one term → recursive tree ✓

$$T(n) = \underbrace{aT\left(\tfrac{n}{b}\right) + f(n)}_{\text{master theorem}}$$

master theorem ✓

Design and analysis of algorithm

↓

now

Recursion tree

↓

masters theorem

THANK - YOU