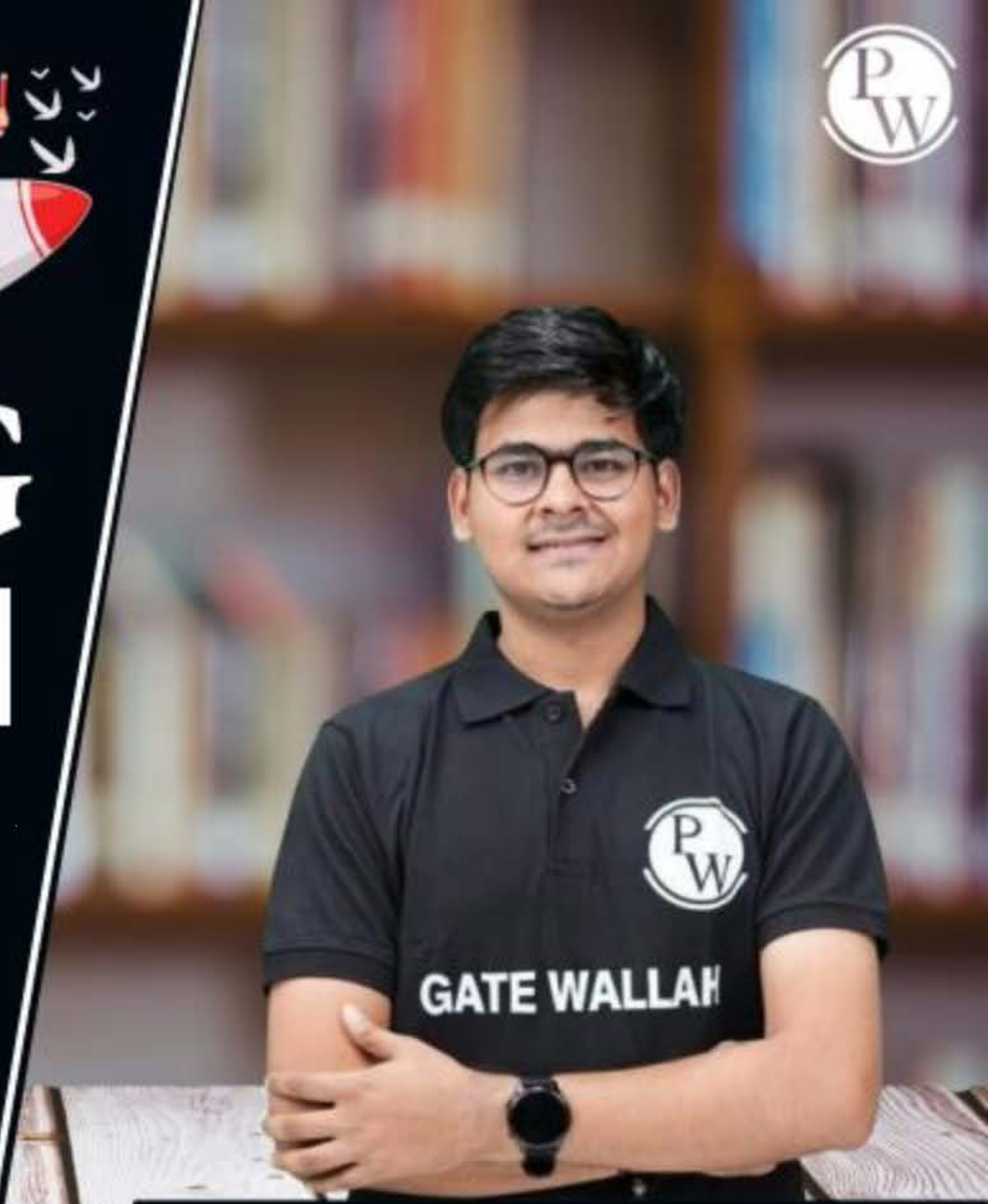
CS & IT ENGINEERING

Algorithms

Divide & Conquer



Lecture No.- 05

By- Aditya sir

Topics to be Covered

Topic

Topic

Masters Mtd for TC analysis

About Aditya Jain sir

1. Appeared for GATE during BTech and secured AIR 60 in GATE in very first attempt - City topper
2. Represented college as the first Google DSC Ambassador.
3. The only student from the batch to secure an internship at Amazon. (9+ CGPA)
4. Had offer from IIT Bombay and IISc Bangalore to join the Masters program
5. Joined IIT Bombay for my 2 year Masters program, specialization in Data Science
6. Published multiple research papers in well known conferences along with the team
7. Received the prestigious excellence in Research award from IIT Bombay for my Masters thesis
8. Completed my Masters with an overall GPA of 9.36/10
9. Joined Dream11 as a Data Scientist
10. Have mentored 12,000+ students & working professions in field of Data Science and Analytics
11. Have been mentoring & teaching GATE aspirants to secure a great rank in limited time
12. Have got around 27.5K followers on Linkedin where I share my insights and guide students and professionals.

HW : Merge Sort

(Q) Merge Sort : 30 sec $\longrightarrow$ $n = 64$ WC

$n = ?$ $t = 6\,min$
WC

$t = 30$ sec,

$n = 64$

WC TC : $O(n \log_2 n)$

$t = C * n \log_2 n$

$30 = C \times 64 \log_2 64$

$30 = C \times 64 \times 6$

$$C = \frac{\cancel{30}}{\cancel{6} \times 64} = \boxed{\frac{5}{64}}$$

Reqd:

$t = 6\ \text{min}, \quad n = ?$

$\quad\; \parallel$

$6 \times 60\ \text{sec}$

$$6 \times 60 = C \times n\log_2 n$$

$$n\log_2 n = \frac{6 \times 60}{C} = \frac{6 \times 60 \times 64}{5} = 9 \times 512$$

$2^9 = 512$

$$n\log_2 n = 512 \times 9$$

$n = 512$

Master's Method / Theorem

Topic : Divide & Conquer

Solving DnC Recurrence

1. Back Substitution method → Universal method
   (i) Value of recurrence
   (ii) TC of recurrence

Eg: Min-max prob :

Value : $\left[\frac{3n}{2} - 2\right]$

TC → O(n)

2. Master's method/Theorem → Shortcut
   (i) Only gives TC but not value
   (ii) Works only in specific cases (3 cases) Symm DnC

3. Recurrence tree approach :
   (i) Asymmetric DnC recurrence

Topic : Divide & Conquer

8.5

Master's method : Only applicable for

9.1/10

$$\boxed{\begin{array}{ll} & T(n) = a * T(n/b) + F(n) \\ \text{Such that} & a \geq 1,\ b > 1,\ F(n) = +ve \end{array}}$$

→ Symmetric DnC

$\frac{\text{Reward}}{\text{efforts}}$

Case 1 : If $F(n)$ is $O\left(n^{(\log_b a - E)}\right)$, for same $E > 0$,

Then $T(n) = \theta\left(n^{\log_b a}\right)$

9.5/10

$\frac{100}{\ }$

$\frac{\text{Reward}}{\text{Risk}}$

Topic : Divide & Conquer

Master's method : Only applicable for

$$T(n) = a * T(n/b) + F(n)$$

Such that $a \geq 1, b > 1, F(n) = +ve$ → Symmetric DnC

Case 1 : If $F(n)$ is $O\left(n^{(\log_b a - E)}\right)$, for same $E > 0$,

Then $\boxed{T(n) = \theta\left(n^{\log_b a}\right)}$

$\log_b a$

$<$

Topic : Divide & Conquer

Case 2 : If F(n) is $\theta\left(n^{\log_b a} * (\log n)^K\right)$ for some K, such that

(a) $K \geq 0$, then $T(n) = \theta\left(n^{\log_b a} * (\log n)^{K+1}\right)$

(b) $K = -1$, then $T(n) = \theta\left(n^{\log_b a} * \log(\log n)\right)$

Topic : Divide & Conquer

Case 3 : If $F(n)$ is $\Omega\left(n^{(\log_b a+E)}\right)$

For some $E > 0$

AND

$a * F(n/b) \leq \delta * f(n)$, For some $\delta < 1$

Then,

$$\boxed{T(n) = \theta(F(n))}$$

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

1) Back-sub $\longrightarrow O(n^2)$

2) MM

Topic : Divide & Conquer

Check Using Back-substitution :

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

$$T(n/2) = 4T(n/2^2) + n/2$$

$$T(n) = 4(4T(n/n^2) + n/2) + n$$

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

$$= 4^2\, T\,(n/2^2) + 3n$$

$$= 4^3\, T\,(n/2^3) + 7n$$

.

.

.

$$\boxed{T(n) = 4^K T\,(n/2^K) + (2^K - 1)n}$$

Now, $n/2^K = 1$

$2^K = n \quad \Rightarrow \quad 4^K = n^2$

$$T(n) = n^2 T(1) + (n - 1)n$$

$$= n^2 * c + n^2 - n$$

$$\boxed{T(n) = \theta\,(n^2)}$$

#Q. $T(n) = 4T(n/2) + n$

Check master's method form :

$$T(n) = a * T(n/b) + F(n)$$

$a = 4, b = 2, F(n) = n \quad \Rightarrow \quad$ Valid

$$\log_b a = \log_2 4 = \log_2 2^2 = 2$$

Topic : Divide & Conquer

Check Case 1 : If $F(n) = O(n^{2-E})$, some $E > 0$

Is $n = O(n^{2-E}), \quad E > 0$

$E = 0.5, 0.1, 0.2, \ldots..$ Yes

$n = O(n^{1.5})$ True

Case 1 satisfied

Hence $\boxed{T(n) = \theta(n^2)}$

2) $T(n) = 2T(n/2) + n\log n$

HW : Back-sub

Topic : Divide & Conquer

#Q. $T(n) = 2T(n/2) + n \log n$

Check : $a = 2, b = 2, F(n) = n \log n$ $\Rightarrow$ Valid

$\log_b a = \log_2 2 = 1$

Check case 1 :

If $F(n) = O\left(n^{\log_b a - E}\right)$, $E > 0$

$n\log n = O\left(n^{1-E}\right)$, some $E > 0$?

Fails

Topic : Divide & Conquer

Check Case 2 :

If $\quad F(n) = \theta\left(n^{\log_b a} * (\log n)^K\right)$

Case 2 : a) $K \geq 0$

$$n\log n = \theta\left(n^1 * (\log n)^K\right) \quad \rightarrow \quad \text{True for } K = 1$$

Case (a) satisfied.

hence, $T(n) = \theta\left(n^1 * (\log n)^{K+1}\right)$, $\quad K = 1$

$$= \theta(n * (\log n)^{1+1})$$

$$\boxed{T(n) = \theta(n * (\log n)^2)}$$

Topic : Divide & Conquer

#Q. $T(n) = T(n/3) + n$

A $T(n) = \theta(n \log n)$

B $T(n) = \theta(n)$

C $T(n) = \theta(n^2)$

D $T(n) = \theta(\log n)$

Validity :

$$a = 1,\ b = 3,\ F(n) = n$$

$$\log_b a = \log_3 1 = 0$$

Topic : Divide & Conquer

$n^{-\varepsilon}$

Case 1 :

If $\quad F(n) = O\left(n^{\log_b a - E}\right), \quad E > 0$

$n = O(n^{0\ -E}), \quad$ some $E > 0$

Fails

Case 2 :

$n = \theta\left(n^{o} * (\log n)^{K}\right), \quad$ some K

(a) $K \geq 0 \quad \rightarrow \quad$ Fails ✗

(b) $K = -1 \quad \rightarrow \quad$ Fails ✗

Topic : Divide & Conquer

Case 3 :

If $\quad F(n) = \Omega\left(n^{\log_b a + E}\right), \quad E > 0$

$n = \Omega\left(n^{0+E}\right), \quad E > 0\ ?$

$n = \Omega(n^{0.1}) \quad E = 0.1, 0.5, 0.2, \ldots\ldots, \ n \geq n^{E}, E > 0$

and

$a * F(n/b) \leq \delta * F(n), \quad$ some $\delta < 1$

$1 * n/3 \leq \cancel{1/3} * n, \quad$ some $\delta < 1$

δ

let take $\boxed{\delta = 1/3}$

$1 * n/3 \leq 1/3 * n \quad \rightarrow \quad$ True

$\rightarrow$ Here case 3 holds true.

Hence $\quad T(n) = \theta(F(n)) \quad \Rightarrow \quad \boxed{T(n) = \theta(n)}$

Topic : Divide & Conquer

#Q. $T(n) = 9T(n/3) + n^{2.5}$

A $T(n) = \theta(n)$

B $T(n) = \theta(n\sqrt{n})$

C $T(n) = \theta(n \log n)$

D $T(n) = \theta(n^2\sqrt{n})$

$n^{2.5} = n^{2+0.5}$

$= n^2\sqrt{n}$

Topic : Divide & Conquer

$a = 9, b = 3, F(n) = n^{2.5}$ → Valid

$\log_b a = \log_3 9 = 2$

$n^2 * \sqrt{n}$

$n^2 * n^{0.5}$

$n^{(2+0.5)}$

$n^{2.5}$

Topic : Divide & Conquer

Case 1 : Is $n^{2.5} = O(n^{2-E})$, Some $E > 0$

$n^{2.5} \leq n^{2-E}$? $\rightarrow$ Not possible

$\rightarrow$ Fails

Case 2 : Is $n^{2.5} = \theta(n^2 * (\log n)^K)$, For some K

(a) $K \geq 0$ $\rightarrow$ Fail ✗

(b) $K = -1$ $\rightarrow$ Fail

$n^{2.5} = n^2(\log n)^K$

$\sqrt{n}$ $(\log n)^K$

$\rightarrow$ Fails ✗

$\cancel{n}^{2.5} = \Theta(\cancel{n^2} \log n^k)$

$\sqrt{n} = \Theta(\log n^k)$

Topic : Divide & Conquer

Case 3: Is $n^{2.5} = \Omega\left(n^{2+E}\right)$ some $E > 0$

$n^{2.5} > n^{2+E}$, $E = 0.1, 0.2, 0.33, \ldots..$

$\rightarrow$ yes ✓

and $a * F\left(\frac{n}{b}\right) \leq \delta * F(n)$ For some $\delta < 1$

$F(n) = n^{2.5}$

$a * \left(\frac{n}{3}\right)^{2.5} \leq \delta * n^{2.5}$

$\frac{9}{3^2\sqrt{3}} * \cancel{n^{2.5}} \leq \delta * \cancel{n^{2.5}}$ $\delta = 1/\sqrt{3}$, $F(n) = n^{2.5}$

Case 3 is satisfied. $\Rightarrow$ $T(n) = \theta\,(F(n)) = \theta\,(n^{2.5})$

Topic : Divide & Conquer

Doubt : $F(n) = n^{2.5}$

$\frac{n^{2.5}}{3}$ ✗

$F\left(\frac{n}{3}\right) \rightarrow \left(\frac{n}{3}\right)^{2.5}$

$n \rightarrow \frac{n}{3}$

Imp: Validate all the DnC recurrences that we have learnt so for using Master's method.

i) $\boxed{\text{Min – Max}}$

ii) Binary Search

iii) Merge Sort

~~iv) Matrix Multiplication~~

~~Hint~~

1. Min-Max Problem :

$T(n) = 2T(n/2) + 2$ $\rightarrow$ $\boxed{\begin{array}{l} \text{Value of Rec} \Rightarrow \left(\frac{3n}{2} - 2\right) \\ TC \rightarrow O(n) \end{array}}$

$\rightarrow$ $a = 2, b = 2,$ $F(n) = 2$ (+ve) $\rightarrow$ Valid

$\log_b a = \log_2 2 = 1$

Case 1 : Is $2 = O(n^{1-E})$, $E > 0$

Yes $E = 0.5, 0.1, 0.2, \ldots\ldots$ True

Hence $T(n) = \theta\left(n^{\log_b a}\right) = \theta(n)$

Topic : Divide & Conquer

2. Binary Search :

$$\boxed{T(n) = T(n/2) + C}$$

$$T(n) = \theta(\log n)$$

→ $a = 1, b = 2, F(n) = C$ → Valid

$$\log_b a = \log_2 1 = 0$$

Check case 1

Is $F(n) = O\left(n^{\log_b a - E}\right)$ Some $E > 0$

$C = O(n^{0-E})$ Some $E > 0$ → Fails

$n^{-E}, E > 0$ → Decreasing function

Topic : Divide & Conquer

Check case 2 :

Is $F(n) = \theta\left(n^{\log_b a} * (\log n)^K\right)$, some K

(a) $K \geq 0$

(b) $K = -1$

$$C = \theta\left(n^0 * (\log n)^K\right) \ ?$$

If $K = 0$, $C = \theta(n^0 * (\log n)^0)$

$$\boxed{C = \theta(1)} \rightarrow \text{True}$$

Hence case 2 (a) holds

$$T(n) = \theta\left(n^{\log_b a} * (\log n)^{K+1}\right)$$

$$= \theta(n^0 * (\log n)^{0+1})$$

$$\boxed{T(n) = \theta\,(\log n)}$$

3. Merge Sort :

$\boxed{T(n) = 2T\,(n/2) + n}$

$TC = \theta\,(n \log n)$

Check $a = 2, b = 2, F(n) = n$ $\rightarrow$ Valid

$\log_b a = \log_2 2 = 1$

Check case 1 :

Is $F(n) = O\left(n^{\log_b a - E}\right)$ some $E > 0$

$n = O(n^{1-E})$, some $E > 0$ $\rightarrow$ Fails

Check case 2 :

Is $F(n) = \theta\left(n^{\log_b a} * (\log n)^K\right)$, Some K

(a) $K \geq 0$

(b) $K = -1$

$K = 0$, $n = \theta\left(n^1 * (\log n)^0\right)$

$\boxed{n = \theta(n)}$ → True

Case holds for $K = 0$

$T(n) = \theta\left(n^{\log_b a} * (\log n)^{K+1}\right)$

$= \theta\left(n^1 * (\log n)^{0+1}\right) = \theta(n\log n)$

Practice Problem for Master's Method :

1. $T(n) = 6T(n/3) + 2^n \log n$
2. $T(n) = 4T(n/2) + \log n$
3. $T(n) = 4T(n/2) + n^2$
4. $T(n) = 2T(n/2) + \sqrt{n}$ → B.S → $\theta(n)$
5. $T(n) = 3T(n/3) + n$
6. $T(n) = 2^n T(n/4) + n$
7. $T(n) = 2T(\sqrt{n}) + \log n$ → Special Case

$$T(n) = 2T(n/2) + \sqrt{n}$$

$$T(n) = \Theta(n^1)$$

$a = 2$

$b = 2$

$$\log_b a = \log_2 2 = 1$$

$F(n) = \sqrt{n}$

$\varepsilon < 0.5$

Is $\sqrt{n} = O(n^{1-\varepsilon})$, some $\varepsilon > 0$

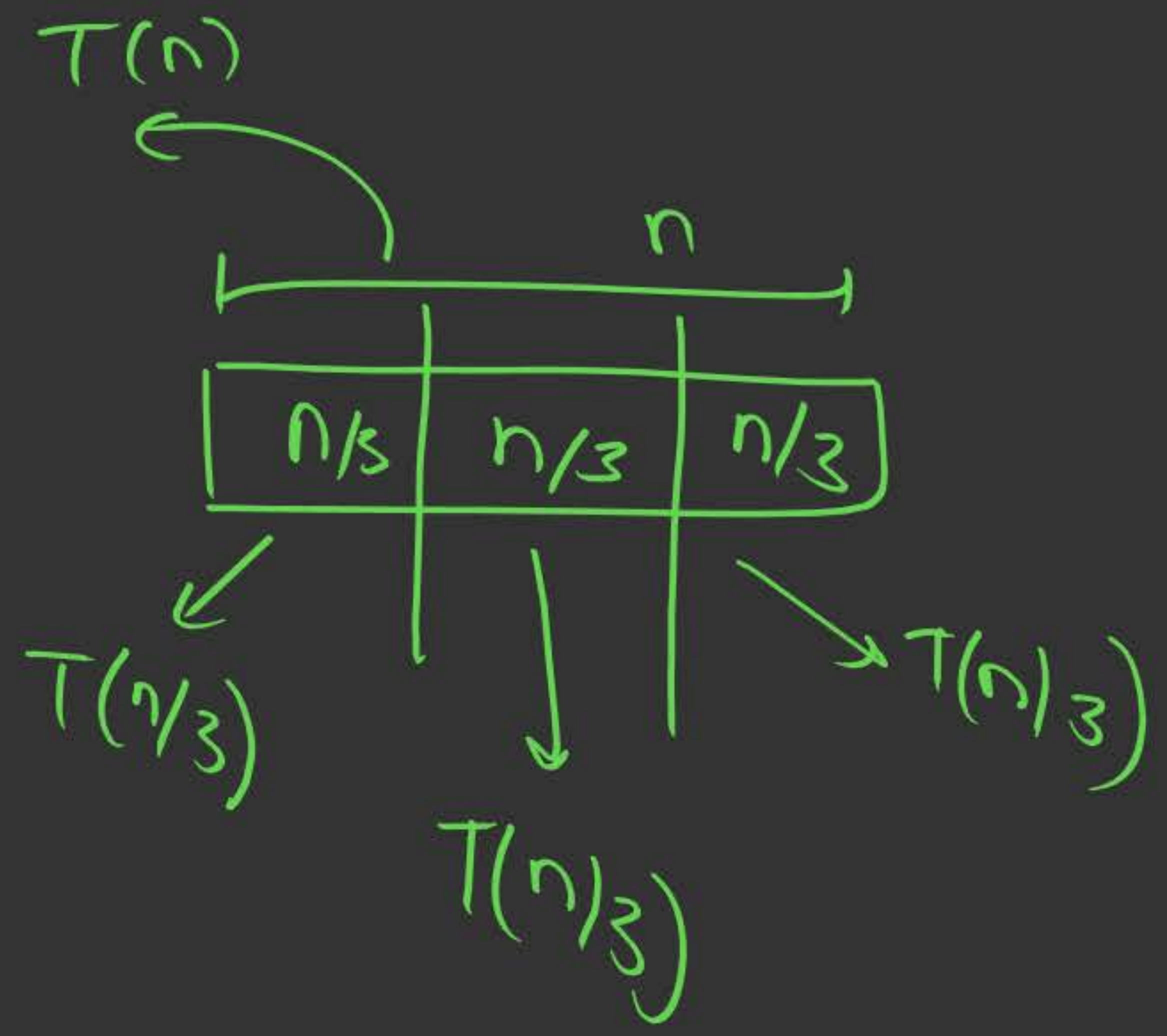
BS

2) $\boxed{T(n) = 3T(n/3) + n}$

$a = 3, \; b = 3, \; F(n) = n$

$\log_b a = \log_3 3 = 1$

Case1: $n = O(n^{1-\varepsilon}), \; \varepsilon > 0$?



fails

Case 2 :-

Is $n = \Theta(n^1 * \log n^k)$

a) $k \geq 0$ $k = 0$

$n = \Theta(n)$ ✓

Hence

$$T(n) = \Theta(n \log n^{0+1}) = \Theta(n \log n)$$

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

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

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

$$\boxed{T(n) = 3^K T(n/3^K) + K * n}$$

$$\frac{n}{3^K} = 1 \rightarrow 3^K = n$$

$$\boxed{K = \log_3 n}$$

$$T(n) = n * b + n \log_3 n$$

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

THANK - YOU