CS & IT ENGINEERING

Algorithms

Divide & Conquer

Lecture No.- 06

By- Aditya sir

Topics to be Covered

Topic

Topic

Adv MM

Matrix Mul

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

$$T(n) = a * \boxed{T(n/b)} + F(n)$$

$a \geq 1$

$b > 1$

$F(n) =$ the function

* **Change of variable mtd :-**

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

let $\boxed{n = 2^k}$

$\left[\sqrt{n} = 2^{k/2}\right]$

$T(2^k) = 2T(2^{k/2}) + \log(2^k)$ — ①

let $T(2^k) = P(k)$

then $P(k/2) = T(2^{k/2})$

$$T(2^k) = 2T(2^{k/2}) + \log 2^k$$

$$P(k) = 2P(k/2) + \log(2^k)$$

$$\boxed{P(k) = 2P(k/2) + k} \text{ — } (2)$$

$a = 2$

$b = 2$

$F(k) = k$

✓

Case 1 : $k = O(k^{1-\varepsilon}), \varepsilon > 0$?

→ Fails

$\log_2 2 = 1$

Case 2:- Is $K = \Theta(K^1 * (\log K)^d)$

a) $\underline{\underline{d \geq 0}}$, $\underline{d = 0}$, $\underline{K = \Theta(K)}$

$P(K) = \Theta(K * (\log K)^{0+1}) = \underline{\Theta(K \log K)}$

$$T(2^k) = \Theta(k \log k)$$

$$T(n) = \Theta(k \log k)$$

$$n = 2^k$$

$$\boxed{k = \log_2 n}$$

$$T(n) = \Theta(\log n * \log(\log n))$$

2) $T(n) = T(\sqrt{n}) + 5$

Let $n = 2^k$

$\sqrt{n} = 2^{k/2}$

$T(2^k) = T(2^{k/2}) + 5$

Let $P(k) = T(2^k)$

$P(k/2) = T(2^{k/2})$

$P(k) = P(k/2) + 5$

$a = 1$

$b = ?$

$f(k) = 5$ ✓

Case 1: $5 = O(k^{0-\varepsilon}), \varepsilon > 0$

Fails

Case 2: Is $5 = \Theta(k^0 * \log k^d)$

a) $d \geq 0$

$d = 0$ $\quad 5 = \Theta(1)$ ✓

$$P(k) = \Theta(k^0 * \log k^{0+1}) = \Theta(\log k)$$

$$P(k) = \Theta(\log k)$$

$$n = 2^k$$

$$\boxed{k = \log_2 n}$$

$$T(2^k) = \Theta(\log k)$$

$$T(n) = \Theta(\log k)$$

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

(Q) $T(n) = 25T(n/5) + \log n$

$T(n) = ?$

A) $\Theta(n)$

B) ~~$\Theta(n^2)$~~

C) $\Theta(\sqrt{n})$

D) $\Theta(\log n)$

$$T(n) = 25T(n/5) + \log n$$

$a = 25$

$b = 5$

$F(n) = \log n$

✓ $\log_b a = \log_5 25 = \underline{\underline{2}}$

Case 1:

$$\log n = O\left(n^{2-\varepsilon}\right), \text{ some } \varepsilon > 0$$

$$\varepsilon = 0.1, 0.3 \checkmark$$

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

Matrix Multiplication Problem

w.r.t. square matrix of size $n \times n$.

Topic : Divide & Conquer

V. Imp

Basics : $A_{n\times n} * B_{n\times n} \rightarrow C_{n\times n}$

In general : $A_{r1\times c1} * B_{r2\times c2} = C_{r1\times c2}$

A and B are multipliable only if $\boxed{c_1 = r_2}$

Eq : Matrix Addition

$r_1 = r_2$
$\& \ c_1 = c_2$

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \qquad B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

$$A + B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}_{2\times 2} + \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}_{2\times 2} = \begin{bmatrix} 1+5 & 2+6 \\ 3+7 & 4+8 \end{bmatrix} = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix}_{2\times 2}$$

$$A_{n\times n} + B_{n\times n} \to C_{n\times n}$$

Time Complexity of Previous Method of Matrix Addition

$A_{n\times n} + B_{n\times n} \Rightarrow$

$$\left\{\begin{array}{l} \text{for } i : 1 \rightarrow n \\ \quad \text{for } j : 1 \rightarrow n \\ \quad\quad C[i, j] = A[i, j] + B[i, j] \end{array}\right.$$

$TC \Rightarrow \boxed{O(n^2)}$ [Matrix Addition]

and

Matrix subtraction

Topic : Divide & Conquer

Matrix Multiplication :

Given two square matrices each $\rightarrow$ n × n

$$\underset{A}{\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}} * \underset{B}{\begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}} = \begin{bmatrix} 1 \times 5 + 2 \times 7 & 1 \times 6 + 2 \times 8 \\ 3 \times 5 + 4 \times 7 & 3 \times 6 + 4 \times 8 \end{bmatrix}_{2\times2}$$

$$= \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

$$\begin{array}{r} 32 \\ 18 \\ \hline 50 \end{array} \qquad \begin{array}{r} 28 \\ 15 \\ \hline 43 \end{array}$$

Code of prev matrix multiplication logic (Non DnC)

$A_{n\times n} * B_{n\times n} = C_{n\times n}$

For i : 1 → n : ⇒ O(1) space

  For j : 1 → n :

    For k : 1 → n

      C[i, j] = A[i, k] * B[k, j]

⇒ $\boxed{TC = O(n^3)}$

Can ~~are~~ we do better using DnC approach ?

Topic : Divide & Conquer

Matrix multiplication in general : (Non DnC)

v. imp

$$\overset{A}{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}} * \overset{B}{\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}} = \overset{C}{\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}}$$

C_{xy}

Then, $t_1 \rightarrow$

		Add	mul
$c_{11} = a_{11} * b_{11} + a_{12} * b_{21}$	$\rightarrow$	1	2
$c_{12} = a_{11} \times b_{12} + a_{12} \times b_{22}$	$\rightarrow$	1,	2
$c_{21} = a_{21} \times b_{11} + a_{22} \times b_{21}$	$\rightarrow$	1,	2
$c_{22} = a_{21} \times b_{12} + a_{22} \times b_{22}$	$\rightarrow$	1,	2

4 -Addition
&
8-Multiplication.

Topic : Divide & Conquer

Eg : Divide & Conquest based approach

$T(n)$

$$\underset{4\times 4}{\begin{array}{cc|cc} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 7 \\ \hline 9 & 2 & 1 & 3 \\ 5 & 7 & 2 & 4 \end{array}} * \underset{4\times 4}{\begin{array}{cc|cc} 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 2 \\ \hline 3 & 1 & 7 & 4 \\ 3 & 2 & 9 & 8 \end{array}} \Rightarrow \underset{4\times 4}{\begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array}}$$

A: A_{11}, A_{12}, A_{21}, A_{22} B: B_{11}, B_{12}, B_{21}, B_{22} C

$\Rightarrow T(n)$

$\Downarrow$

8 Multiplication

4 Addition

$16\times 16 \rightarrow 4\times 4$

$4\times 4 \rightarrow 2\times 2$

Small problem

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} * \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

Sub-problem

Where,

2×2 $T(n/2)$ $T(n/2)$

	Add	Mul
$C_{11} = A_{11} * B_{11} + A_{12} * B_{21}$ - (1)	1,	2
$C_{12} = A_{11} * B_{12} + A_{12} * B_{22}$ - (2)	1,	2
$C_{21} = A_{21} * B_{11} + A_{22} * B_{21}$ - (3)	1,	2
$C_{22} = A_{21} * B_{12} + A_{22} \times B_{22}$ - (4)	1,	2

4 sub-matrix addition & 8 sub-matrix Mul

Subproblem → Divide

$A, B \rightarrow n \times n$

Topic : Divide & Conquer

Divide :

$n \times n \rightarrow \left(\frac{n}{2} \times \frac{n}{2}\right)$ Subproblem

$\downarrow$ Original problem $\downarrow$ $n \times n$

$\downarrow$ $\frac{n}{2} \times \frac{n}{2}$

$T(n) = 8 * T(n/2) + O(n^2) \rightarrow$ gate

Topic : Divide & Conquer

Let T(n) represent the time complexing to multiply two square matrices A & B each of size $n \times n$. (square matrix) → Recurrance for DnC based matrix multiplication.

* $T(n) = C, \quad n \leq 2$

$T(n) = 8\,T(n/2) + b * n^2, \quad n > 2, \; b > 0$

$T(n/2) = 8T\,(n/2^2) + b * n^2/4$

$T(n) = 8\left[8T\left(\frac{n}{2^2}\right) + \frac{bn^2}{4}\right] + b * n^2$

$= 8^2T\left(\frac{n}{2^2}\right) + 2bn^2 + bn^2$

$= 8^2T\left(\frac{n}{2^2}\right) + 3bn^2 \qquad \text{.... (2)}$

$2^k = n$

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

$2^{3k} = n^3$

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

Topic : Divide & Conquer

General Term :

$8^k = n^3$

$$T(n) = 8^k T\left(\frac{n}{2^k}\right) + (2^k - 1) * bn^2 \qquad \dots(2)$$

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

$2^k \rightarrow n$

$8^k \rightarrow n^3$

$$T(n) = n^3 T(1) + (n-1) * bn^2$$

$$= n^3 * c + (n-1) * bn^2$$

$$= n^3 * c + bn^3 - bn^2 \Rightarrow \boxed{O(n^3)}$$

TC after using this simple DnC based approach is also $O(n^3)$.

Topic : Divide & Conquer

Observations :

1. In DnC based matrix multiplication, there are 8 subproblems (sub-matrix multiplication), involved in getting C_{11}, C_{12}, C_{21}, C_{22}.
2. Time complexing will only be reduced, if the no. of sub-matrix multiplications are reduced from 8 to a smaller value.

$$\Downarrow$$

"STRASSEN" → Research

Logic :

$$a * b = \underbrace{a + a + \ldots\ldots + a}_{b \text{ times}}$$

Topic : Divide & Conquer

Strassen's Matrix Multiplication

$$P = (A_{11} + A_{22})(B_{11} + B_{22})$$
$$Q = (A_{21} + A_{22})B_{11}$$
$$R = A_{11}(B_{12} - B_{22})$$
$$S = A_{22}(B_{21} - B_{11})$$
$$T = (A_{11} + A_{12})B_{22}$$
$$U = (A_{21} - A_{11})(B_{11} + B_{12})$$
$$V = (A_{12} - A_{22})(B_{21} + B_{22})$$

$$C_{11} = P + S - T + V$$
$$C_{12} = R + T$$
$$C_{21} = Q + S$$
$$C_{22} = P + R - Q + U$$

Result of Strassen's research on matrix multiplications.

$\{A_{n\times n} * B_{n\times n} \rightarrow C_{n\times n}\} \rightarrow$ Original problem

$A_{ij}, B_{ij}, C_{ij} \Rightarrow \frac{n}{2} \times \frac{n}{2}$

Additionally created/used sub-matrices

$P, Q, R, S, T, V, U \Rightarrow \left(\frac{n}{2} \times \frac{n}{2}\right)$

Topic : Divide & Conquer

Let T(n) represents the TC for the matrix multiplication using Strassen's approach :

$$T(n) = c, \quad n \le 2$$

$$T(n) = 7 \times T\left(\frac{n}{2}\right) + bn^2, n > 2$$

Strassen's Recurrence

$a = 7,\ b = 2,\ f(n) = n^2$

$$T(n/2) = 7 * T\left(\frac{n}{2^2}\right) + b\left(\frac{n}{2}\right)^2$$

C1: $n^2 = O(n^{\log_2 7 - \varepsilon})$, $\varepsilon > 0$

$$T(n) = 7\left[7T\left(\frac{n}{2^2}\right) + \frac{bn^2}{2}\right] + bn^2$$

$n^2 = O(n^{2.81 - \varepsilon})$ ✓

$$= 7^2 T\left(\frac{n}{2^2}\right) + \frac{7bn^2}{4} + bn^2$$

$T(n) = O(n^{2.81})$

Topic : Divide & Conquer

General Term :

$$T(n) = 7^k T\left(\frac{n}{2^k}\right) + bn^2 * \left(\sum_{i=0}^{k-1} \left(\frac{7}{4}\right)^i\right)$$

$$\text{GP} : \sum_{i=1}^{n} x^i < x^{n+1}$$

$$T(n) = 7^k T\left(\frac{n}{2^k}\right) + bn^2 * \left(\frac{7}{4}\right)^k$$

$$= 7^k T(1) + bn^2 * \left(\frac{7}{4}\right)^k$$

$$= 7^k * c + bn^2 * \frac{7^k}{4k}$$

$$\sum_{i=1}^{3} = 2^1 + 2^2 + 2^3$$

$$= 2 + 4 + 8$$

$$= 14 < 2^k$$

$$= 14 < 16$$

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

$2^k = n$ $4^k = n^2$

$k = \log_2 n$

Topic : Divide & Conquer

$$T(n) = 7^k * c + bn^2 * \frac{7^k}{n^2}$$

$$T(n) = 7^k * c + 7^{12} * b$$

$$= 7^k (b + c)$$

$$T(n) = d * 7^k$$

$$T(n) = O(7^k)$$

$$\boxed{T(n) = O(n^{2.81})}$$

$$n^3 = n^{2.81}$$

Let $b + c = d$ (constant)

$$k = \log_2 n$$

$$7k = 7^{(\log_2 n)}$$

$$= n^{\log_2 7}$$

$$= n^{2.81}$$

$$\log_2 7 = 2.81$$

Topic : Divide & Conquer

Complexity Analysis of matrix mul (square) : Time complexity summary

1.	Non – DnC	$\Rightarrow$	$O(n^3)$
2.	Simple DnC	$\Rightarrow$	$O(n^3)$
3.	Strassen's based DnC	$\Rightarrow$	$O(n^{2.81})$

Topic : Divide & Conquer

Space Complexity :

1. Non – DnC $\Rightarrow$ $O(1)$ $\rightarrow$ Constant
2. Simple DnC $\Rightarrow$ Recurrsion stack $\rightarrow$ $O(\log_2 n)$
3. Strassen's DnC $\Rightarrow$ $O(\log n + n^2) = \underline{O(n^2)}$ $\rightarrow$ P, Q, R, S, R, U, V

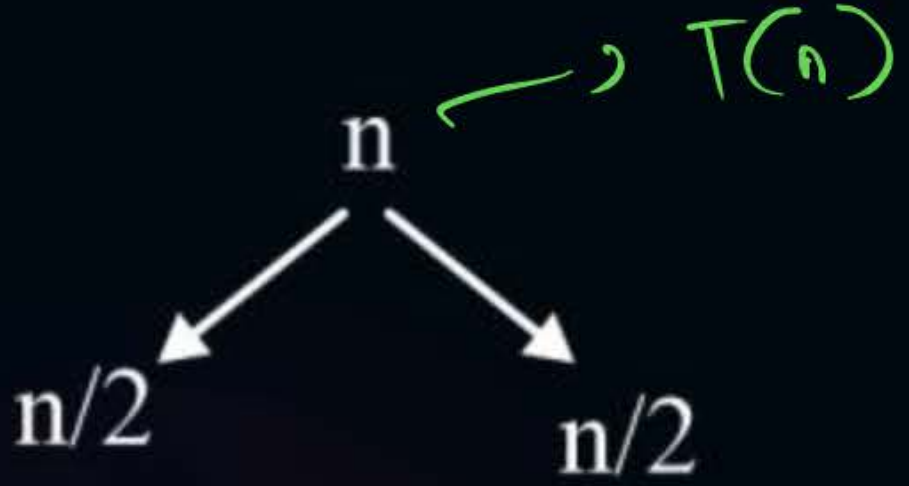
TC: $O(n^3) = O(n^{2.81})$

Long Integer Multi

Long Integer Multiplication

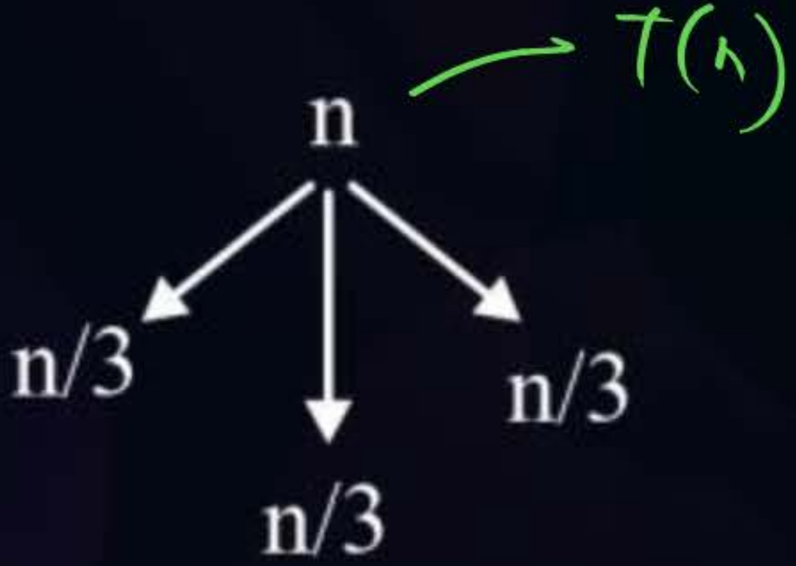
Idea:-

2-way split



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

3-way split



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

$a = 30$

$b = 50$

$a * b = ?$ $\underline{\underline{1500}}$

LIM

$a = $ | 2 | 3 | 5 | 1 | 2 | 9 | 7 | 3 | 1 | 2 | 9 | 8 |

$b = $ | 3 | 2 | 5 | 7 | 8 | 3 | 2 | 9 | 7 | 3 | 8 | 2 |

$$
\begin{array}{r}
10 \\
* \quad 15 \\
\hline
50 \\
10* \\
\hline
150 \\
\hline
\end{array}
$$

3 approaches

Topic : Recurrence after 3-way split

(1) Simple DnC $\Rightarrow T(n) = 9T(n/3) + n$

$n^{\log_3^9} = n^2$

(2) AK optimization DnC $\Rightarrow$ $T(n) = 8T(n/3) + n$

$n^{\log_3^8} = n^{1.89}$

(3) T & C Optimization DnC $\Rightarrow 5T(n/3) + n$

$n^{1.46}$

Topic : Summary to use in GATE

Summary to use in GATE

(1) 2-way split: DnC → 4, AK → 3, T & C → 3

(2) 3-way split: DnC → 9, AK → 8, T & C → 5

(3) 4-way split: DnC → 16, AK → 15, T & C → 7

k-way k^2 (k^2-1) $(2k-1)$

Recurrence equation,

(1) DnC → 16T (n/4) + bn, n > 1

(2) AK opti. → 15T (n/4) + bn, n > 1

(3) T&C opt.→ 7T (n/4) + bn, n > 1

→ 4-way split

Topic : Summary to use in GATE

Generalised for k-way split

DnC → k^2 Ak → $(k^2 - 1)$ T&C → $(2k - 1)$

The generalized time complexity recurrences for k-way split:

(1) DnC → $T(n) = k^2\, T(n/k) + bn$

(2) Anatoly Karatsuba optimization.(AK optimization) → $T(n) = (k^2 - 1)\, T(n/k) + bn$

(3) T&C opt.→ $T(n) = (2k - 1)T(n/k) + bn$

5-way split

1) DnC $\longrightarrow 25T(n/5) + bn$

2) AK $\longrightarrow 24T(n/5) + bn$

3) T&C $\longrightarrow 9T(n/5) + bn$

Topic : Divide & Conquer

#Q. The running time of an algorithm is represented by the following recurrence relation:

$$T(n) = \begin{cases} n & n \le 3 \\ T\left(\frac{n}{3}\right) + cn & otherwise \end{cases}$$

Which of the following represents the time complexity of the algorithm?

A. θ(n)

B. θ(n logn)

C. $\theta(n^2)$

D. $\theta(n^2 \log n)$

$$\boxed{\begin{array}{l} n=1,\ 1 \\ n=2,\ 2 \\ n=3,\ 3 \end{array}}$$

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

$$\left.\begin{array}{l} a=1 \\ b=3 \\ f(n)=n \end{array}\right] \checkmark$$

$$\log_b a$$

$$= \log_3 1 = \underline{\underline{0}}$$

C1 :- $cn = O(n^{0-\epsilon})$, $\epsilon > 0$?

fails

C2 : $cn = \Theta(n^0 * \log n^k)$

a) $k \geq 0$ ✗ $cn = \Theta(\log n^k)$

b) $k = -1$ ✗

C3:

$cn = \Omega(n^{0+\varepsilon}), \varepsilon > 0?$

&

$a * f(n/b) \le \delta * f(n)$

True

$1 * f(n/3) \le \delta * f(n)$

$\frac{c\cancel{n}}{3} \le \delta * c\cancel{n} \Rightarrow \delta \ge 1/3$

$T(n) = \Theta(f(n)) = \Theta(cn) = \underline{\Theta(n)}$

#Q. QS on an unsorted list of size n, $(n/4)^{th}$ smallest element is selected as pivot each time, then the time complexity recurrence is ____.

Quick Sort

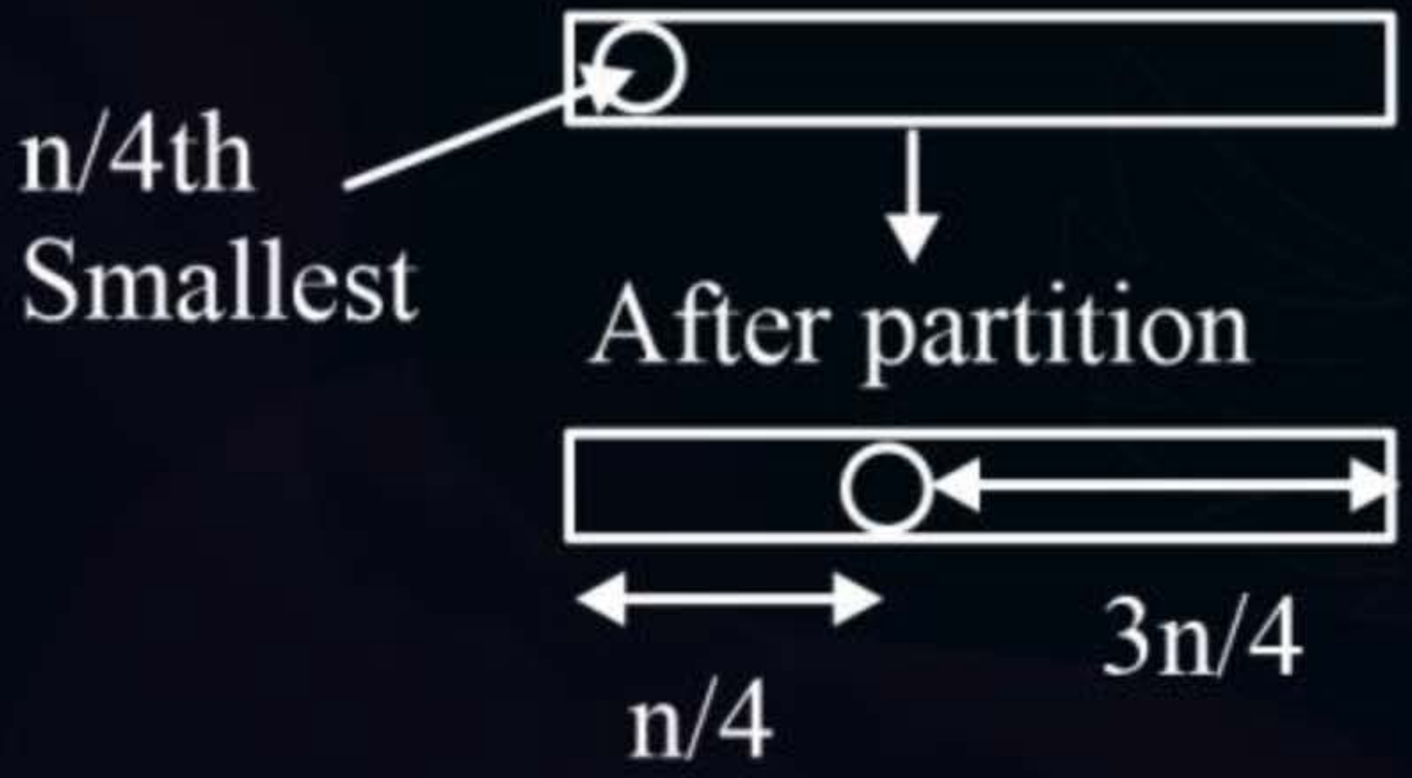
HW

This is available in $O(n)$ time

Topic : Divide & Conquer

$$T(n) = T(n/4) + T(3n/4) + b \times n$$

$\Rightarrow \quad TC \rightarrow O(n \log n)$



THANK - YOU