CS & IT ENGINEERING

Algorithms

Divide & Conquer

Lecture No.- 05

By- Dr. Khaleel Khan Sir

Recap of Previous Lecture

Topic Matrix Multiplication

Topic Master Theorem

Topic

Topic

Topic

Topics to be Covered

- **Topic** Master Theorem
- **Topic** Long Integer Multiplication (LIM)
- **Topic**
- **Topic**
- **Topic**

MCQ

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

$a=1;\ b=3;\ \log_3 1=0$

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

n is it $O(n^{0-\epsilon})$ ✗
n is it $\Theta(n^0 \cdot \log^k n)$ ✗
n is it $\Omega(n^{0+\epsilon})$ ✓

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

[GATE-2009: 2M]

(A) $\Theta(n)$ ✓

(B) $\Theta(n \log n)$

(C) $\Theta(n^2)$

(D) $\Theta(n^2 \log n)$

MCQ

Which one of the following correctly determines the solution of the recurrence relation with T(1) = 1?

$$T(n) = 2T\left(\frac{n}{2}\right) + \log n$$

[GATE-2014: 1M]

A) $\Theta(n)$

B) $\Theta(n\log n)$

C) $\Theta(n^2)$

D) $\Theta(\log n)$

$a = 2$
$b = 2$

$\log n$ is it $O(n^{1-\epsilon})$

$\epsilon = 0.5$

$O(\sqrt{n})$

MCQ

For Constants a $\geq$ 1 and b > 1, consider the following recurrence defined on the non-negative integers:

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$$

Which one of the following options is correct about the recurrence T(n)?

[GATE-2021: 2M]

(A) If $f(n)$ is $\Theta\left(n^{\log_b(a)}\right)$ then T$(n)$is $\Theta\left(n^{\log_b(a)}\right)$

(B) If $f(n)$is $O\left(n^{\log_b(a)-e}\right)$for some $e > 0$, then T(n) is$\Theta\left(n^{\log_b(a)}\right)$.

(C) If $f(n)$is $\frac{n}{\log_2(n)}$, then T$(n)$is $\Theta(n\,log_2(n))$.

(D) If $f(n)$is $n\log_2(n)$, then T$(n)$is $\Theta\left(\text{n}\log_2(n)\right)$.

MCQ

For parameters a and b, both of which are $\omega(1)$, $T(n) = T(n^{1/a}) + 1$, and $T(b) = 1$. Then $T(n)$ is [GATE-2020: 1M]

(A) $\Theta(\log_2 \log_2 n)$

(B) $\Theta(\log_a \log_b n)$

(C) $\Theta(\log_b \log_a n)$

(D) $\Theta(\log_{ab} n)$

Back Subst

$$T(n) = T(n^{1/a}) + 1 \quad -(1)$$

$$T(n^{1/a}) = T(n^{1/a^2}) + 1 \quad -(2)$$

$$T(n) = T(n^{1/a^2}) + 2 \quad -(3)$$

$$= T(n^{1/a^k}) + k \quad -(4)$$

$$= T(1) + (\log_a \log_b n)$$

$$n^{1/a^k} = b$$

$$\frac{1}{a^k} \cdot \log_2 n = \log_2 b$$

$$a^k = \frac{\log_2 n}{\log_2 b} = \log_b n$$

$$a^k = \log_b n$$

$$k \cdot \log_2 a = \log_2 (\log_b n)$$

$$\boxed{k = \frac{\log_2(\log_b n)}{\log_2 a} = \log_a \log_b n}$$

Topic : Algorithms

Consider the following recurrence relation

H/W

$T(n) = T(n-1) + \lfloor \sqrt{n} \rfloor$

$T(1) = 1$

$$T(n+1) = T(n) + \left\lfloor \sqrt{n+1} \right\rfloor \textit{ for all } n \geq 1$$

The value of $T(m^2)$ for $m \geq 1$ is

A. $\frac{m}{6}(21m - 39) + 4$

B. $\frac{m}{6}\left(4m^2 - 3m + 5\right)$

C. $\frac{m}{2}\left(3m^{2.5} - 11m + 20\right) - 5$

D. $\frac{m}{6}\left(5m^3 - 34m^2 + 137m - 104\right) - \frac{5}{6}$

H/W

When n = 2^{2k} for some $k \geq 0$, then recurrence relation

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

Evaluates to:

A. $\sqrt{(n)}(\log n + 1)$

B. $\sqrt{(n)}\log n$

C. $\sqrt{(n)}\log\sqrt{(n)}$

D. $n\log\sqrt{(n)}$

$a = \sqrt{2}$, $b = 2$ $\log_2 \sqrt{2} = 1/2$

Case I : $\sqrt{n}$ is it $O(n^{1/2 - \epsilon})$ ×

Case II : $\sqrt{n}$ is it $\Theta(n^{1/2} \cdot \log^k n)$

$\therefore T(n)$ is $\Theta(\sqrt{n} \cdot \log n)$ ✓ $k = 0$

1. $T(n) = 2^n T(n/2) + n$: In-Admissible Case,

   $2^n \rightarrow$ Constant

2. $T(n) = 0.5\, T(n/2) + n^2$: " " "

   $0.5 \rightarrow a \geq 1$

1) $T(n) = 2 \cdot T(\sqrt{n}) + \log n$ 		$b=1$, $b>1$

Changing the variables;

Let $n = 2^K \Rightarrow K = \log n$

$$T(2^K) = 2 \cdot T(2^{K/2}) + K \quad \text{— (1)}$$

Let $T(2^K) = S(K)$

$T(2^{K/2}) = S(K/2)$

$T(n) = 2T(n/2) + n \rightarrow \Theta(n \cdot \log n)$

$$\boxed{S(K) = 2 \cdot S(K/2) + K} \quad \text{— (3)}$$

$\downarrow$

$\Theta(K \cdot \log K) \Rightarrow \boxed{\Theta(\log n * \log \log n)}$ ✓

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

$n = 2^k$

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

let $\underline{T(2^k)} = \underline{S(k)}$

$$\boxed{S(k) = S(k/2) + 1 \quad - (2)}$$

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

$\Theta(\log k) \Rightarrow \Theta(\log \log n)$

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

MCQ

H/W : (DS + Algo)

An algorithm performs $(\log N)^{1/2}$ find operations, N insert operation, $(\log N)^{1/2}$ delete operations, and $(\log N)^{1/2}$ decrease-key operations on a set of data items with keys drawn from a linearly ordered set. For a delete operation, a pointer is provided to the record that must be deleted. For the decrease-key operation, a pointer is provided to the record that has its key decreased. Which one of the following data structures is the most suited for the algorithm to use, if the goal is to achieve the best total asymptotic complexity considering all the operations?

[GATE-2015: 2M]

(A) Unsorted array

(B) Min – heap

(C) Sorted array

(D) Sorted doubly linked list

6) Long-Integer Multiplication (LIM)

→ Given Two integers, say 'u' & 'v';

int u, v; → regular Integers (2B)

↓

Max : 32767

a) u+v;
b) u*v;

} O(1)

→ Long int u, v; → 4B/8B → 8/10 digit no

→ No H/w Soln

→ S/w Solution? Repr. the Integers in Array form

$u = 12345871 \Longrightarrow$

	1	2	3	4	5	6	7	8
u :	1	2	3	4	5	8	7	6
v :	8	9	6	4	3	2	1	9

9 1 0 9

→ Long int's of 'n' digits each are repr. in an array of Size 'n'

int u[n], v[n], c[n];

a) $u \pm v \Longrightarrow O(1)$? "No" $\rightarrow O(n)$ ✓

$$\boxed{\begin{array}{l}\text{for } i \leftarrow 1 \text{ to } n \\ \quad c[i] = u[i] \pm v[i];\end{array}}$$

with Possible Carry

b) $u * v = c$;

$O(n^2)$ ✓

$$\left.\begin{array}{l}\text{for } i \leftarrow 1 \text{ to } n \\ \quad \text{for } j \leftarrow 1 \text{ to } n \\ \quad\quad = \\ \quad\quad = \\ \quad\quad =\end{array}\right\}$$

$$\begin{array}{r} 1\,2\,3 \\ \times\ 4\,5\,6 \\ \hline 7\,3\,8 \\ 6\,1\,5\ \ \\ 4\,9\,2\ \ \ \ \\ \hline \\ \hline \end{array}$$

School method

DandC - Method for LIM

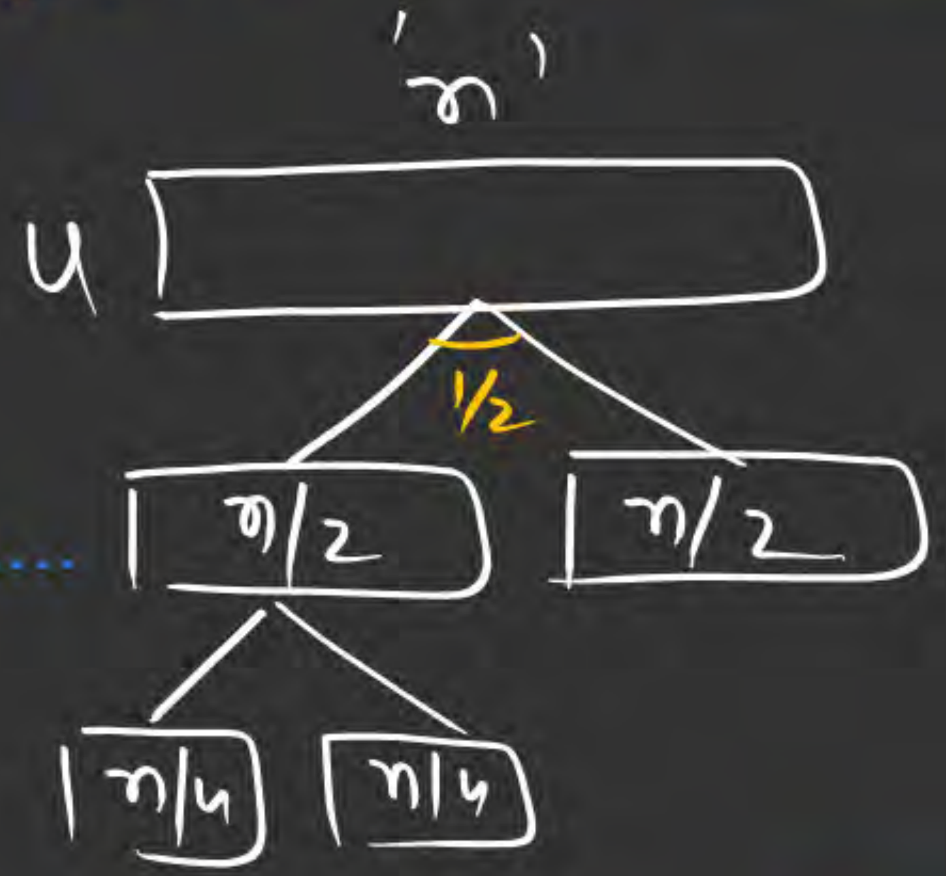
u, v: Long int's of 'n' digits each,

$u = 1234$; $v = 5678$

$= 12 \times 10^2 + 34$ $\qquad = 56 \times 10^2 + 78$

$1 \times 10^1 + 2$ $\qquad 3 \times 10^1 + 4$

$1234 / 10^2$ $\qquad 1234 \% 10^2 = 34$



w, x, y, z : n/2 digit no's

Let $m = \lfloor n/2 \rfloor$

$u = (w \times 10^m + x)$; $v = (y \times 10^m + z)$

$w = u / 10^m$; $x = u \% 10^m$

$y = v / 10^m$; $z = v \% 10^m$

$$\underline{u \cdot v} = (w \times 10^m + x) \cdot (y \times 10^m + z)$$

n

$$T(n) \qquad = \boxed{\underbrace{w \cdot y}_{n/2} \times 10^{2m} + \left[\underbrace{(w \cdot z)}_{n/2} + \underbrace{(x \cdot y)}_{n/2}\right] \times 10^m + \underbrace{x \cdot z}_{n/2}} \text{ --- } (1)$$

Let $T(n)$ repr. Time Complexity of Multiplying Two long int's of n-digits each,

$$(u\cdot v)_n = (w\cdot y)_{n/2} \times 10^{2n} + (w\cdot z + x\cdot y)_{n/2} \times 10^{n} + (x\cdot z)_{n/2} \quad - (1)$$

D and C

$$\boxed{\begin{aligned} T(n) &= 4\cdot T(n/2) + bn, & n > 1 \quad - (1) \\ &= c, & n = 1 \quad - (2) \end{aligned}} \Rightarrow \boxed{T(n) = \Theta(n^2)}$$

$O(n^{\log_2 4})$

Anatoly Karatsuba (A.K) optimization:

Let t : $n/2$ digit no;

$$\boxed{t = (w+x)\cdot(y+z)}$$

$$= w\cdot y + (w\cdot z + x\cdot y) + x\cdot z$$

$$(w\cdot z + x\cdot y) = [t - (w\cdot y + x\cdot z)]$$

$\rightarrow$ Let

$$\left.\begin{aligned} w\cdot y &= Pr_1 \\ x\cdot z &= Pr_2 \\ t &= Pr_3 \end{aligned}\right\}$$

(A.K)

$$\boxed{(u\cdot v)_n = Pr_1 \times 10^{2n} + Pr_3 - (Pr_1 + Pr_2)\times 10^{n} + Pr_2} \quad - (2)$$

For A·K optimization

$$\boxed{\begin{aligned} T(n) &= 3 \cdot T(n/2) + b \cdot n, \quad n > 1 \\ &= c, \quad n = 1 \end{aligned}}$$

Case I:

$$\Theta\left(n^{\log_2 3}\right) = \Theta\left(n^{1.58}\right)$$

Toom & Cook optimization (TC):

→ 3-way-Split:

$(u\cdot v)_n = 9\,(x\cdot y)_{n/3}$ — (1)

1) $T(n) = \underline{9\cdot T(n/3)} + n$ — (2) (Dandc)

→ $\Theta(n^2)$

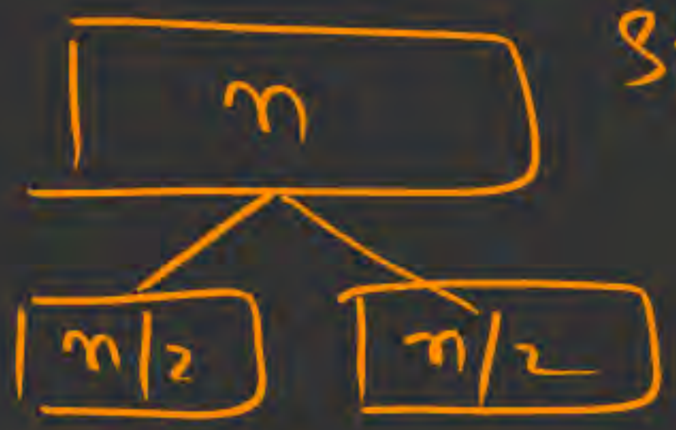
2) A·K optimiz : $\underline{8}\,T(n/3) + n$ — (3)

→ $\Theta(n^{\log_3 8}) = \Theta(n^{1.89})$

: $x\cdot T(n/3) + n$ — (4)

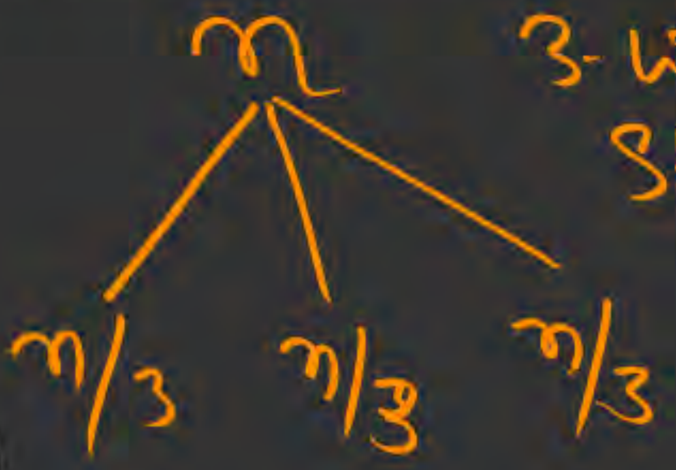
→ $\Theta(n^{\log_3 x}) < \Theta(n^{1.58})$

2-way split



Dandc : $4\cdot T(n/2)$

3-way split



$\log_3 x < 1.58$

$x < 3^{1.58} < 3\sqrt{3} = 5$

If $(x = 5)$ then T.C will be better

→ They solved system of linear Eqn's to get $(u \cdot v)_n = (w \cdot x)_{n/3}$ with 5

Subproblems

$T(n) = 5 \cdot T(n/3) + n$ — 1

$\rightarrow \Theta(n^{\log_3 5}) = \Theta(n^{1.46})$ ✓

4-way Split:

1) D and C : $T(n) = 16 \cdot T(n/4) + n \Rightarrow \Theta(n^2)$

2) A · K : $T(n) = 15 \cdot T(n/4) + n \Rightarrow \Theta(n^{1.95})$

3) T · C : $T(n) = x \cdot T(n/4) + n \Rightarrow \Theta(n^{\log_4 x}) < \Theta(n^{1.46})$

$$x = 4^{1.46} \sim 7$$

$$T(n) = 7 \cdot T(n/4) + n$$

$$\hookrightarrow \Theta(n^{\log_4 7}) = \Theta(n^{1.40})$$

1) 2-way Split, DC: 4 ; A·K : 3 ; T·C = 3
2) 3-way Split ; DC : 9 ; A·K : 8 , T·C = 5
3) 4-way " . D.C : 16 ; A·K : 15 ; T·C = 7
4) K-way " : D·C : K^2 ; A·K : (K^2-1) ; T C = $(2K-1)$

Generalized Eq's of Time Complexities for K-way Split

① D and C : $T(n) = K^2 \cdot T(n/K) + bn$ — ①

② A K : $T(n) = (K^2-1) \cdot T(n/K) + bn$ — ②

③ T·C : $T(n) = (2K-1) \cdot T(n/K) + bn$ — ③

II. GREEDY METHOD (G.M)

- G.M is a design strategy used for solving problems, whose solutions are viewed (seen) as a result of making a set of decisions;
- These set of decisions based on G.M are made in a step-wise (step-by-step) manner;
- At each step, out of available options, greedily select that option that satisfies the given criteria/conditions/objectivity;

THANK - YOU