→ Recurrence Relation:

For recursive functions, recurrence relation can be established & can be used to analyze the time complexity of recursive functions.

Example:
To calculate factorial of a number using recursive function

Time complexity

```
algo fact (n)                                   T(n)
{
    if (n == 0 || n == 1)    } ---------------  1
    return 1                 } (base condition)

    else
    return  n * fact (n-1)                      T(n-1)
                 (recursive call)
                                                1
}
```

– ∴ Recurrence relation formation

Let $fact(n) \rightarrow T(n)$
∴ $fact(n-1) \rightarrow T(n-1)$

$$\therefore \boxed{\begin{aligned} T(n) &= T(n-1) + 1, & n > 1 \\ &= 1, & n = 0 \text{ or } n = 1 \end{aligned}}$$

For second term in $\underbrace{T(n-1)}_{\text{term 1}} + \underbrace{1}_{\text{term 2}}$, always asymptotic representation is used

Methods of solving recurrence relation

There are 3 methods of solving recurrence relation

1) substitution method } can be applied to
2) recursion tree method } all recurrence relation
3) master's theorem — cannot be applied to all recurrence relations

decreasing function

① solve $T(n) = T(n-1) + 1 \quad , n > 1$

$= 1 \quad , n = 0$ or $n = 1$

② using substitution method

$T(n) = \underline{T(n-1)} + 1$ ——— ①

∴ Put $n = n-1$ in eqⁿ ①.

∴ $T(n-1) = T((n-1)-1) + 1$

∴ $T(n-1) = \underline{T(n-2)} + 1$ ——— ②

Put $n = n-2$ in eqⁿ ①

∴ $T(n-2) = T((n-2)-1) + 1$

$T(n-2) = \underline{T(n-3)} + 1$ ——— ③

Put $n = n-3$ in eqⁿ ①

∴ $T(n-3) = T((n-3)-1) + 1$

$T(n-3) = T(n-4) + 1$ ——— ④

Put eq^n (2) in eq^n (1)

$\therefore T(n) = T(n-1) + 1$

$\therefore T(n) = [T(n-2)+1] + 1$

$\therefore T(n) = T(n-2) + 2$ —— (5)

Put eq^n (3) in eq^n (5)

$\therefore T(n) = [T(n-3)+1] + 2$

$\therefore T(n) = T(n-3) + 3$ —— (6)

Put eq^n (4) in eq^n (6)

$\therefore T(n) = [T(n-4)+1] + 3$

$\therefore T(n) = T(n-4) + 4$ —— (7)

⋮

~~Assume~~

$\therefore T(n) = T(\underbrace{n-k}_{T(1)}) + k$ —— (8)

Assume $n-k=1$

$\therefore k = n-1$

substitute value of 'k' in eq^n (8)

$\therefore T(n) = T(n-(n-1)) + (n-1)$

$\therefore T(n) = T(\cancel{n}-\cancel{n}+1) + (n-1)$

$\therefore T(n) = T(1) + (n-1)$ —— (9)

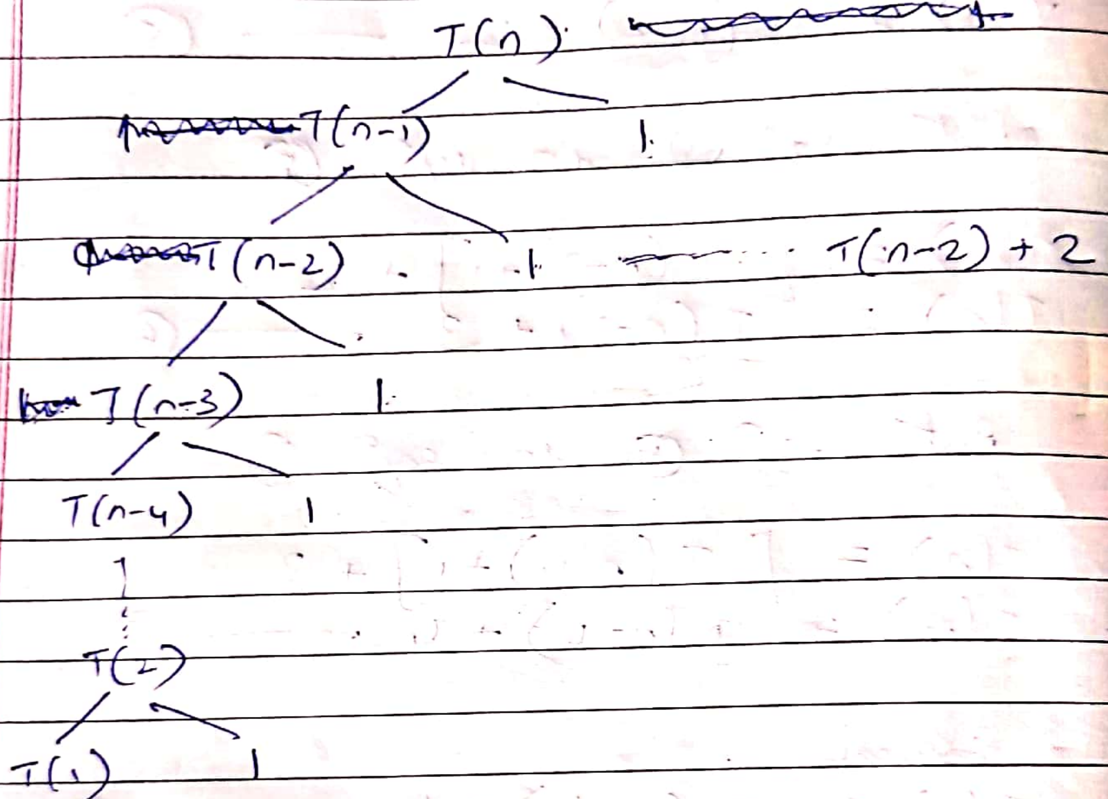
From given recursion relation, $T(1) = 1$, put this value in eq^n (9)

$T(n) = \cancel{1} + n - \cancel{1}$

$\therefore T(n) = n$

$\therefore \boxed{T(n) = O(n)}$ ← Time complexity

① using recursion tree method
calculate the total running time at each level of tree.



~~T(n) = 1 + 1 + 1 + 1 + ... = k times add~~

$\therefore T(n) = T(1) + [1 + 1 + 1 + \dots + 1] =$ add 1 k

$\therefore T(n) = 1 + \sum_{i=1}^{k} 1$

$\therefore T(n) = 1 + k$

Put $k = n-1$.

* Time complexity = sum of work done at each level

$$T(n) = T(n-k) + [1+1+1+\dots k \text{ times}]$$

$$\therefore T(n) = T(n-k) + \sum_{i=1}^{k} 1$$

$$\therefore T(n) = T(n-k) + k$$

Put $n-k=1$, $\therefore k = n-1$

$$\therefore T(n) = T(n-(n-1)) + (n-1)$$

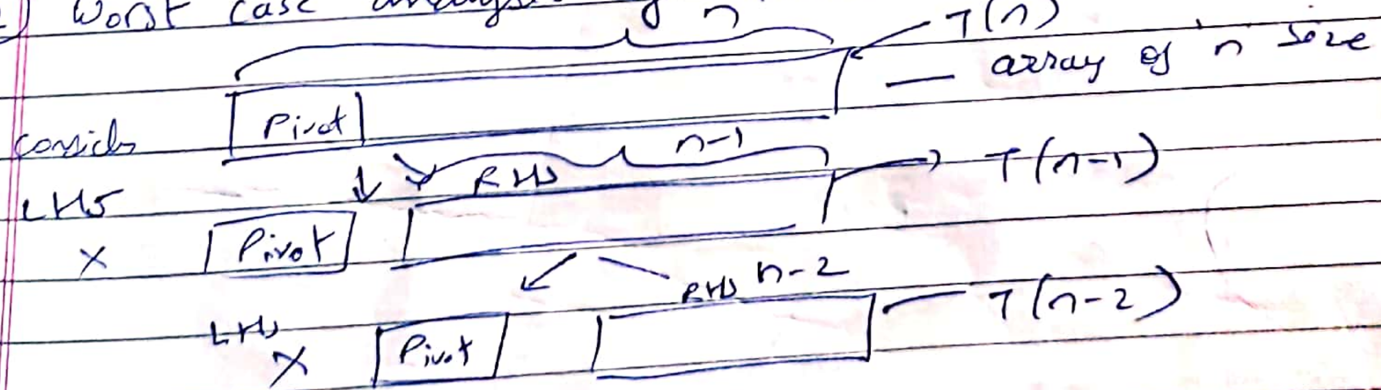
$$\therefore T(n) = T(n-n+1) + (n-1)$$

$$\therefore T(n) = T(1) + (n-1)$$

$$\therefore T(n) = \cancel{1} + n - \cancel{1}$$

$$\therefore \boxed{T(n) = O(n)}$$ — Time complexity

(2) Worst case analysis of Quick sort algorithm.



$$\therefore \boxed{\begin{aligned} T(n) &= T(n-1) + n, & n > 1 \\ &= 1, & n = 1 \end{aligned}}$$

here $T(n-1) \rightarrow$ time to partition $(n-1)$ elements

$n \rightarrow$ time required for comparisons of pivot with other elements.

(ii) using substitution method

$T(n) = T(n-1) + n$ —— ①

Put $n = n-1$ in ①

$\therefore T(n-1) = T((n-1)-1) + n-1$

$\therefore T(n-1) = T(n-2) + (n-1)$ —— ②

Put $n = n-2$ in ①

$\therefore T(n-2) = T((n-2)-1) + (n-2)$

$\therefore T(n-2) = T(n-3) + (n-2)$ —— ③

Put $n = n-3$ in ①

$\therefore T(n-3) = T((n-3)-1) + (n-3)$

$\therefore T(n-3) = T(n-4) + (n-3)$ —— ④

Put ② in ①

$\therefore T(n) = [T(n-2) + (n-1)] + n$

$\therefore T(n) = T(n-2) + (n-1) + n$ —— ⑤

Put ③ in ⑤

$\therefore T(n) = [T(n-3) + (n-2)] + (n-1) + n$

$\therefore T(n) = T(n-3) + (n-2) + (n-1) + n$ —— ⑥

Put ④ in ⑥

$\therefore T(n) = [T(n-4) + (n-3)] + (n-2) + (n-1) + n$

$\therefore T(n) = T(n-4) + (n-3) + (n-2) + (n-1) + n \ldots$

$\vdots$

$\therefore T(n) = T(1) + [(n-3) + (n-2) + (n-1) + n]$

$\therefore T(n) = 1 + \dots + (n-3) + (n-2) + (n-1)$.

$\therefore T(n) = 1 + 2 + 3 + \dots + n$

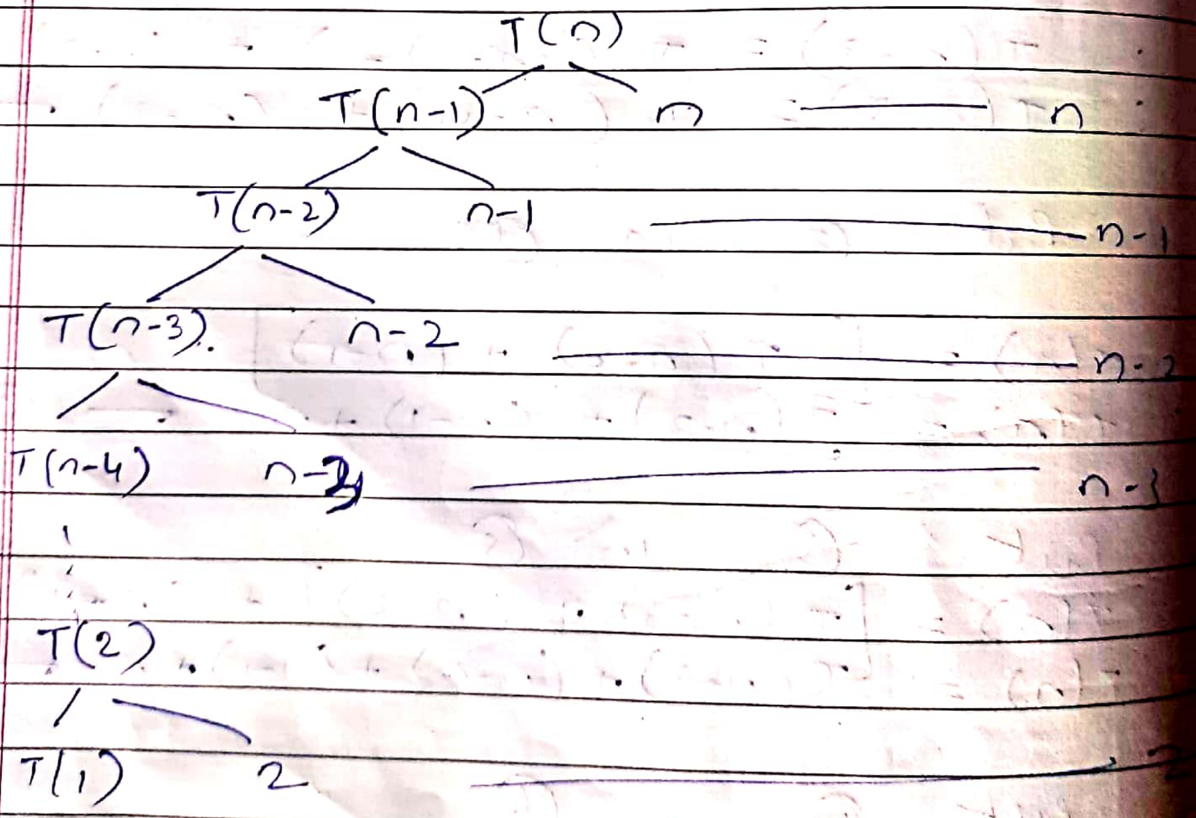
$\therefore T(n) = \frac{n(n+1)}{2}$

$\therefore T(n) = \frac{n^2}{2} + \frac{n}{2}$

$\therefore$ ~~T(n) =~~ Fastest growing term is $\frac{n^2}{2}$

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

(b) recursion tree method



Time complexity = Sum of work done at each level

$\therefore T(n) = T(1) + [2 + \dots + (n-3) + (n-2) + (n-1) + n]$

$\therefore T(n) = 1 + 2 + \dots + (n-3) + (n-2) + (n-1) + n$

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

$\therefore T(n) = \frac{n^2}{2} + \frac{n}{2}$

$\therefore$ Fastest growing term is $\frac{n^2}{2}$

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

(3) ~~solve~~: $T(n) = T(\frac{n}{2}) + T(\frac{n}{2}) + n, \quad n > 1$

$= 1, \quad n = 1$

$T(n) = T(\frac{n}{2}) + T(\frac{n}{2}) + n, \quad n > 1$

$\therefore T(n) = 2T(\frac{n}{2}) + n, \quad n > 1$

dividing function

$$\therefore \boxed{\begin{aligned} T(n) &= 2T(\tfrac{n}{2}) + n, & n > 1 \\ &= 1, & n = 1 \end{aligned}}$$

~~For merge sort, $T(\frac{n}{2})$ = algo mergesort (A,~~

This represents the recurrence relation of
- merge sort algorithm
- best case for quick sort algorithm

Scanned with CamScanner

For merge sort

```
algo mergesort (A, l, h)                 Time complexity
{                                        —— T(n)
    if (l < h)
    {
        mid = (l + h)/2
        mergesort (A, l, mid)            —— T(n/2)
        mergesort (A, mid+1, h)          —— T(n/2)
        merge (A, l, mid, h)             —— n
    }
}
```

$$T(n) = \begin{cases} 2T\left(\frac{n}{2}\right) + n, & n > 1 \\ 1, & n = 1 \end{cases}$$

For quick sort

```
algo quicksort (A, l, h)                 Time complexity
{                                        —— T(n)
    if (l < h)
    {
        pos = partition (A, l, h)        —— n
        quicksort (A, l, pos-1)          —— T(n/2)
        quicksort (A, pos+1, h)          —— T(n/2)
    }
}
```

For best case

solve,

a) using substitution method

$$T(n) = 2T\left(\frac{n}{2}\right) + n \quad \text{—— (1)}$$

Put $n = \frac{n}{2}$ in (1)

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

$$\therefore T\left(\frac{n}{2}\right) = 2\left[T\left(\frac{n}{4}\right)\right] + \frac{n}{2}$$

$$\therefore T\left(\frac{n}{2}\right) = 2T\left(\frac{n}{4}\right) + \frac{n}{2} \quad \text{—— (2)}$$

Put $n = \frac{n}{4}$ in (1)

$$\therefore T\left(\frac{n}{4}\right) = 2T\left(\frac{n}{8}\right) + \frac{n}{4} \quad \text{—— (3)}$$

Put $n = \frac{n}{8}$ in (1)

$$\therefore T\left(\frac{n}{8}\right) = 2T\left(\frac{n}{16}\right) + \frac{n}{8} \quad \text{—— (4)}$$

Put (2) in (1)

$$\therefore T(n) = 2\left[2T\left(\frac{n}{4}\right) + \frac{n}{2}\right] + n$$

$$\therefore T(n) = 4T\left(\frac{n}{4}\right) + n + n$$

$$\therefore T(n) = 4T\left(\frac{n}{4}\right) + 2n \quad \text{—— (5)}$$

Put (3) in (5)

$\therefore T(n) = 4\left[2T\left(\frac{n}{8}\right) + \frac{n}{4}\right] + 2n$

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

$\therefore T(n) = 8T\left(\frac{n}{8}\right) + 3n$ —— (6)

Put (4) in (6)

$\therefore T(n) = 8\left[2T\left(\frac{n}{16}\right) + \frac{n}{8}\right] + 3n$

$\therefore T(n) = 16T\left(\frac{n}{16}\right) + n + 3n$

$\therefore T(n) = 16T\left(\frac{n}{16}\right) + 4n$ —— (7)

⋮

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

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

$\therefore 2^k = n$

$\therefore k = \log_2 n$

} Put in (8)

$\therefore T(n) = nT(1) + (\log_2 n).n$

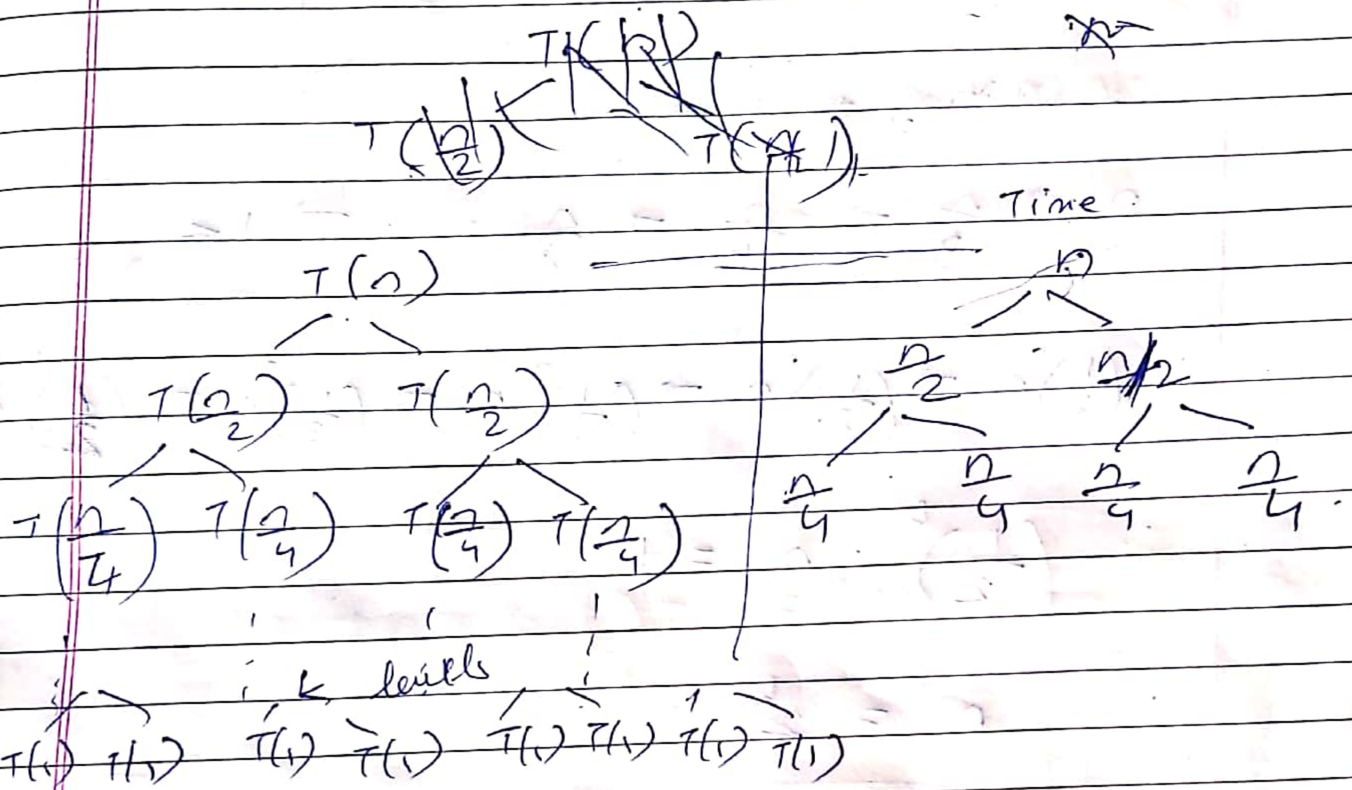
$\therefore T(n) = n*1 + n.\log_2 n$

$\therefore T(n) = n + \underbrace{n.\log_2 n}$ → fastest growing term

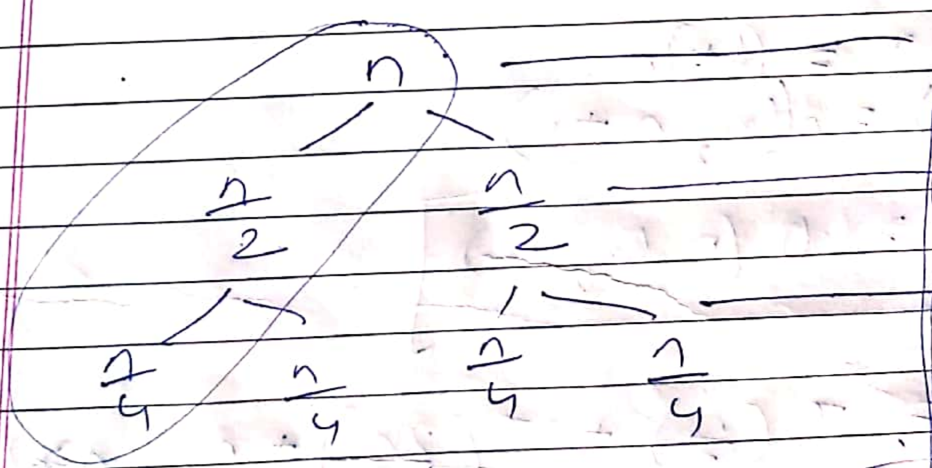
$\therefore \boxed{T(n) = O(n\log_2 n)}$

Scanned with CamScanner

b) using recursion tree method



:. work done at each level



Sum at each level

n

$\frac{n}{2} + \frac{n}{2} = n$

$\frac{n}{4} + \frac{n}{4} + \frac{n}{4} + \frac{n}{4} = n$

upto k+1 levels, where k = height of tree

$\therefore$ $\boxed{\text{Total sum} = (k+1)n}$

:. ~~To~~

To calculate height i.e. k, consider longest path

Here both, left hand side and right hand side represent longest path

Lets consider left hand side path

~~$\frac{n}{2}$ $\frac{n}{4}$~~

$$n \rightarrow \frac{n}{2} \rightarrow \frac{n}{4} \rightarrow \frac{n}{8} \; \text{-----} \rightarrow 1$$

$$\therefore \left(\frac{1}{2}\right)^0 n \rightarrow \left(\frac{1}{2}\right)^1 n \rightarrow \left(\frac{1}{2}\right)^2 n \rightarrow \left(\frac{1}{2}\right)^3 n \rightarrow \left(\frac{1}{2}\right)^k$$

$$\therefore \left(\frac{1}{2}\right)^k \cdot n = 1$$

$$\therefore n = 2^k$$

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

$$\therefore T(n) = (k+1)n$$

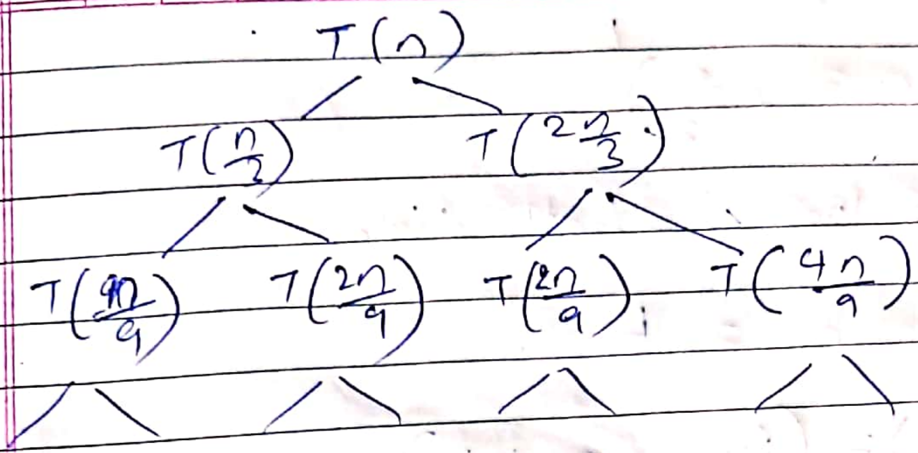
$$\therefore T(n) = n(\log_2 n + 1)$$

$$\therefore \boxed{T(n) = O(n \log_2 n)}$$

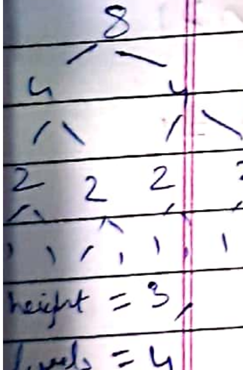
(4) $$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n \quad , n > 1$$

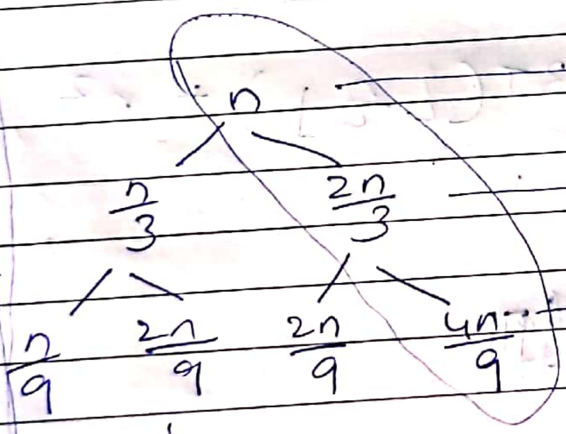
$$= 1 \quad , n = 1$$

Solve using recursion tree method



work done at each level





sum at each level

n

$\frac{n}{3} + \frac{2n}{3} = n$

$\frac{n}{9} + \frac{2n}{9} + \frac{2n}{9} + \frac{4n}{9} = n$

upto $(k+1)$ levels, where k is the height of tree

$\therefore T(n) = (k+1) \cdot n$

consider the longest path in the recursion tree

$\therefore n \longrightarrow \frac{2n}{3} \longrightarrow \frac{4n}{9} \cdots \longrightarrow 1$

$\therefore \left(\frac{2}{3}\right)^0 n \longrightarrow \left(\frac{2}{3}\right)^1 n \longrightarrow \left(\frac{2}{3}\right)^2 n \cdots \longrightarrow \boxed{\left(\frac{2}{3}\right)^k n}$

Assume $\left(\frac{2}{3}\right)^k \cdot n = 1$

Scanned with CamScanner

$\therefore n = \left(\frac{3}{2}\right)^k$

taking log on both side

$\log_{3/2} n = k$

$\therefore T(n) = (\log_{3/2} n + 1)\, n$

$\therefore \boxed{T(n) = O(n \cdot \log_{3/2} n)}$

③ solve $T(n) = 3T\left(\left\lfloor \frac{n}{4} \right\rfloor\right) + n^2 , \; n > 1$

$= 1 , \; n = 1$

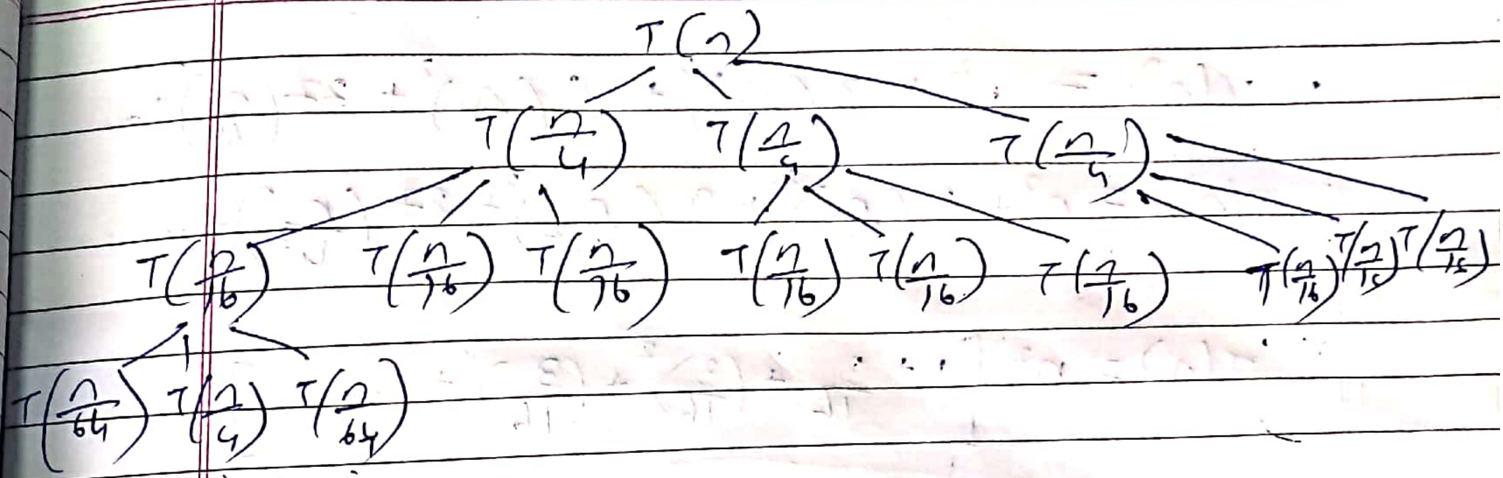
$\lceil \; \rceil$ = ceil

$\lfloor \; \rfloor$ = floor

consider $T(n) = 3T\left(\frac{n}{4}\right) + n^2 , \; n > 1$

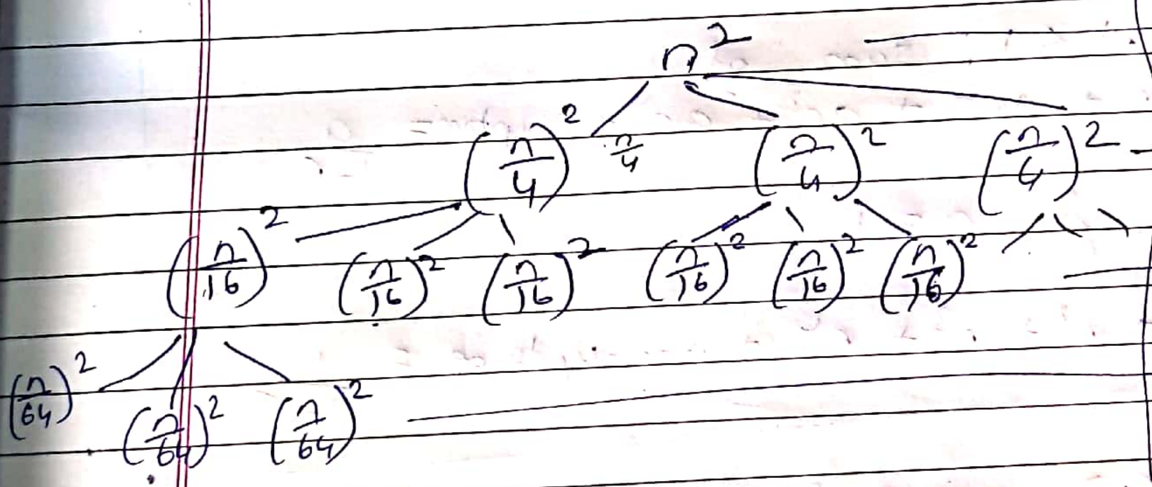
$= 1 , \; n = 1$

using recursion tree.



upto (k+1) levels.

∴ work done at each level.



sum of work done at each level

n^2

$3\left(\frac{n}{4}\right)^2$

$9\left(\frac{n}{16}\right)^2$

$27\left(\frac{n}{64}\right)^2$

Time complexity = sum of work at each level

$\therefore T(n) = n^2 + 3\left(\frac{n}{4}\right)^2 + 9\left(\frac{n}{16}\right)^2 + 27\left(\frac{n}{64}\right)^2 + \cdots$

$\therefore \cancel{T(n) = n^2 + \frac{3}{4}}$

$\therefore \cancel{T(n) = n^2\left[1 + \frac{3}{16} + 9\right.}$

$\therefore T(n) = n^2 + \frac{3n^2}{16} + \frac{9n^2}{16^2} + \frac{27n^2}{64^2} + \cdots$

$\therefore T(n) = n^2 + \frac{3n^2}{4^2} + 9n^2$

$$\therefore T(n) = n^2 + 3\left(\frac{n}{4}\right)^2 + 9\left(\frac{n}{16}\right)^2 + 27\left(\frac{n}{64}\right)^2 \ldots$$

$$\therefore T(n) = n^2 + 3\left(\frac{n}{4}\right)^2 + 9\left(\frac{n}{4^2}\right)^2 + 27\left(\frac{n}{4^3}\right)^2 + \ldots$$

$$\therefore T(n) = n^2 \left[\underbrace{1 + \frac{3}{16} + \left(\frac{3}{16}\right)^2 + \left(\frac{3}{16}\right)^3 + \ldots}\right]$$

geometric Progression with common ratio$(r) = \frac{3}{16}$

$\because$ for $|r| < 1$, sum of

$a + ar + ar^2 + ar^3 + ar^4 + \ldots \text{upto} \infty = \frac{a}{1-r}$

~~for a~~ for $a = 1$,

$1 + r + r^2 + r^3 + r^4 + \ldots \text{upto} \infty = \frac{1}{1-r}$

$$\therefore T(n) = n^2 \left[\frac{1}{1 - \frac{3}{16}}\right] \quad \because \left[\frac{3}{16}\right] < 1$$

$$\therefore T(n) = n^2 \left[\frac{1}{\frac{13}{16}}\right]$$

$$\therefore T(n) = n^2 \underbrace{\left(\frac{16}{13}\right)}_{\text{constant}}$$

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

Scanned with CamScanner

master's Theorem:

- It is used to solve recurrence relation.
- It cannot be applied to all recurrence relations

If $T(n) = a\,T(n/b) + f(n)$

where $f(n) = \theta(n^k \log^p n)$ and

$a \geq 1$, $b > 1$, $k \geq 0$ and p is real number

#

case 1:

if $\log_b a > k$, then $T(n) = \theta(n^{\log_b a})$

case 2:

if $\log_b a = k$.

a) if $p > -1$, then $T(n) = \theta(n^k \log^{p+1} n)$

b) if $p = -1$, then $T(n) = \theta(n^k \log \log n)$

c) if $p < -1$, then $T(n) = \theta(n^k)$

case 3:

if $\log_b a < k$.

a) if $p \geq 0$, then $T(n) = \theta(n^k \log^p n)$

b) if $p < 0$, then $T(n) = \theta(n^k)$

Scanned with CamScanner

Solving examples using master's theorem

① $T(n) = 9T\left(\frac{n}{3}\right) + n$.

$a = 9, \quad b = 3, \quad k = 1, \quad p = 0$.

$\because a \geq 1, \; b > 1, \; k \geq 0$ & p is real
master's theorem can be applied

$\therefore \log_b a = \log_3 9 = 2$

$\because \log_b a > k$.

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

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

② $T(n) = 2T\left(\frac{n}{2}\right) + n$ (merge sort & ~~quick sort~~)

$a = 2, \quad b = 2, \quad k = 1, \quad p = 0$.

$\because a \geq 1, \; b > 1, \; k \geq 0$ & p is real,
master's theorem can be applied.

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

$\because \log_b a = k$ and $p > -1$.

$T(n) = \theta(n^k \log^{p+1} n)$

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

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

Scanned with CamScanner

③ $T(n) = 2^n T\left(\frac{2n}{5}\right) + n$.

$a = 2^n$, $b = \frac{5}{2}$, $k = 1$, p is real.

$\because$ a is not a constant, master's theorem cannot be applied.

④ $T(n) = 0.5\, T\left(\frac{n}{2}\right) + n$.

$\because a < 1$, master's theorem cannot be applied.

Strassen's matrix multiplication.

(I) normal matrix multiplication.

```
algo mul (a, b, n)
{
    for (i=0; i<=n-1; i++)
    {
        for (j=0; j<=n-1; j++)
        {
            c[i][j] = 0
            for (k=0; k<=n-1; k++)
            {
                c[i][j] = c[i][j] +
                    a[i][k] * b[k][j]
            }
        }
    }
}
```

n^3

$\therefore \boxed{T(n) = O(n^3)}$.

eg: $\begin{bmatrix} 4 & 6 & 9 \\ 7 & 8 & 3 \end{bmatrix} \begin{bmatrix} 1 & 7 \\ 2 & 4 \\ 3 & 5 \end{bmatrix} = \begin{bmatrix} - & - & - \\ - & & - \end{bmatrix}$

matrix a: 2×3, $p \; q$

matrix b: 3×3, $r \times s$

matrix c: $p \times s$, 2×3

DATE

(2) using divide & conquer technique

- assume following.
  1) can be applied to square matrix
  2) n is a power of 2 i.e. $n = 2^k$

consider 2×2 matrix multiplication:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

where

$$\left.\begin{aligned} c_{11} &= a_{11} \times b_{11} + a_{12} \times b_{21} \\ c_{12} &= a_{11} \times b_{12} + a_{12} \times b_{22} \\ c_{21} &= a_{21} \times b_{11} + a_{22} \times b_{21} \\ c_{22} &= a_{21} \times b_{12} + a_{22} \times b_{22} \end{aligned}\right\}$$

It consists of 8 multiplication operations & 4 addition operations.

using divide & conquer, consider $n = 4$

$$\left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ \hline a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{array}\right]_{4\times4} \times \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix}_{4\times4} =$$

$$\begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{bmatrix}_{4\times4}$$

Divide each matrix into 4 submatrices $\left(\frac{n}{2}\right)$ such as matrix A divided as

matrix $(A_{11}) = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$, matrix $(A_{12}) = \begin{bmatrix} a_{13} & a_{14} \\ a_{23} & a_{24} \end{bmatrix}$

matrix $(A_{21}) = \begin{bmatrix} a_{31} & a_{32} \\ a_{41} & a_{42} \end{bmatrix}$, matrix $(A_{22}) = \begin{bmatrix} a_{33} & a_{34} \\ a_{43} & a_{44} \end{bmatrix}$

||ly divide matrix B & matrix C.

then, the 3 matrices A, B, C can be represented as:

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

$$\therefore C_{11} = A_{11} \times B_{11} + A_{12} \times B_{21}$$
$$C_{12} = A_{11} \times B_{12} + A_{12} \times B_{22}$$
$$C_{21} = A_{21} \times B_{11} + A_{22} \times B_{21}$$
$$C_{22} = A_{21} \times B_{12} + A_{22} \times B_{22}$$

It consists of 8 matrix multiplication equations & 4 matrix addition.

$$\therefore T(n) = 8T\left(\frac{n}{2}\right) + 4n^2$$

since asymptotic notation is used for second Term.

$$T(n) = 8T\left(\frac{n}{2}\right) + n^2, \quad n > 2$$
$$= 1 \qquad n \leq 2$$

using master's theorem,

$a = 8, \quad b = 2, \quad k = 2, \quad p = 0$

since $a \geq 1, b > 1, k \geq 0$ & p is real

master's theorem can be applied

$\therefore \log_b a = \log_2 8 = 3$

since $\log_b a > k$

$\therefore T(n) = \Theta(n^{\log_b a})$

$\therefore \boxed{T(n) = \Theta(n^3)}$

(3) using strassen's matrix multiplication

consider.

$$\begin{array}{cc} \text{A} & \text{B} \\ \end{array} \text{ row} \quad \quad \begin{array}{cc} \text{E} & \text{F} \end{array} \text{ col}$$

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

$$\begin{array}{cc} \text{C} & \text{D} \end{array} \quad \quad \begin{array}{cc} \text{G} & \text{H} \end{array}$$

where $A_{11}, A_{12}, A_{21}, A_{22}$ are submatrices of matrix A. $||^{ly}$ for matrix B and matrix C

using strassen's method, the values of ~~mat~~ sub matrices C_{11}, C_{12}, C_{21} & C_{22} can be calculated using 7 matrix ~~multiplica~~ multiplication operations, ~~B~~ as follows.

			shortcut	
P_1	=	$A * (F - H)$	A	row +
P_2	=	$(A+B) * H$	H	col +
P_3	=	$(C+D) * E$	E	
P_4	=	$D * (G - E)$	D	
P_5	=	$(A+D) * (E+H)$	diagonal	
P_6	=	$(B-D) * (G+H)$	last CR	
P_7	=	$(A-C) * (E+F)$	First CR	

$\therefore C_{11} = P_4 + P_5 + P_6 - P_2$
$C_{12} = P_1 + P_2$
$C_{21} = P_3 + P_4$.
$C_{22} = P1 - P3 + P5 - P7$

$\therefore$ using strassen's method

$$T(n) = 7T\left(\frac{n}{2}\right) + n^2; \quad n > 2$$
$$= 1, \quad n \leq 2$$

using master's theorem.

$$T(n) = \Theta(n^{\log_b a})$$

$$\therefore T(n) = \Theta(n^{\log_2 7})$$

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