

i) You need  $\log_2 N$  steps to reduce N to 1 by dividing by 2 every time.

$$\textcircled{2} \quad \begin{cases} [3, 10] \Rightarrow 3, 4, 5, 6, 7, 8, 9, 10 \\ (3, 10) \Rightarrow 4, 5, 6, 7, 8, 9 \end{cases}$$

→ Maths

---

guru Class

$$[a, b] \Rightarrow b - a + 1$$

$$\textcircled{3} \quad \frac{AP}{=} \quad \begin{array}{ccccccc} 3, & 6, & 9, & 12, & 15 & \dots \\ \diagdown & \diagdown & \diagdown & \diagdown & \diagdown \\ 3 & 3 & 3 & 3 & 3 \end{array}$$

$$S_n = \frac{n}{2} (2a + (n-1)d)$$

- n → no. of terms
- a → first term
- d → common diff

$$\textcircled{4} \quad \underline{\text{GP}}$$

$2, \quad 6, \quad 18, \quad 54 \dots$

$$S_n = \frac{a(r^n - 1)}{r - 1} \quad (r \neq 1)$$

$a \rightarrow$  first term  
 $r \rightarrow$  common ratio  
 $n \rightarrow$  no. of terms

⑤

$$\log_a a^x = \underline{x}$$

$$\left. \begin{array}{l} \log_2^{32} = 5 \\ \log_2 2^5 \rightarrow 5 \end{array} \right\}$$

$$\log_{10}^{100} \rightarrow 2$$

$$\log_{10}^{(10)^2} \rightarrow 2$$

①

void func( int N ) {

S = 0;

$\Rightarrow \left[ \begin{array}{l} \text{for } (i=1; i \leq N; i++) \{ \\ \quad S = S + 1; \end{array} \right] \quad \begin{array}{l} i = 1 \rightarrow N \\ [1, N] \rightarrow \mathbb{N} \end{array}$

}

No of iterations = N

TC : O(N)

②

void fn( int N, int M ) {

$\left. \begin{array}{l} \text{for } (i=1; i \leq N; i++) \{ \\ \quad \text{if } (i \% 2 == 0) \\ \quad \quad \text{Print ('even');} \end{array} \right\} \quad N$

)  $O(N)$

$\left. \begin{array}{l} \text{for } (j=1; j \leq M; j++) \{ \\ \quad \text{if } (j \% 2 == 1) \\ \quad \quad \text{Print ('odd');} \end{array} \right\} \quad M$

) J

Q3

```
void fn( int N) {
    s=0;
    for (i=1; i<=N; i=i+2) {
        s = s+i;
    }
}
```

$$\frac{N+1}{2} = \frac{N}{2} + \frac{1}{2}$$

odd

$$\begin{array}{c} i \\ \hline 1 \\ 3 \\ 5 \\ 7 \\ \vdots \end{array}$$

No of iteration =  $N/2$

$$\boxed{\frac{[i, N]}{2} \Rightarrow \frac{(N+i)}{2}}$$

$$N = 7 \rightarrow 8$$

1, 3, 5, 7

TC : O(N)

Q4

```
void fn( int N) {
    s=0;
    for (i=0; i<=100; i++) {
        s = s+i+i^2;
    }
}
```

$$[0, 100] \Rightarrow 101$$

No of iteration = 101

TC : O(1)

Constant Time

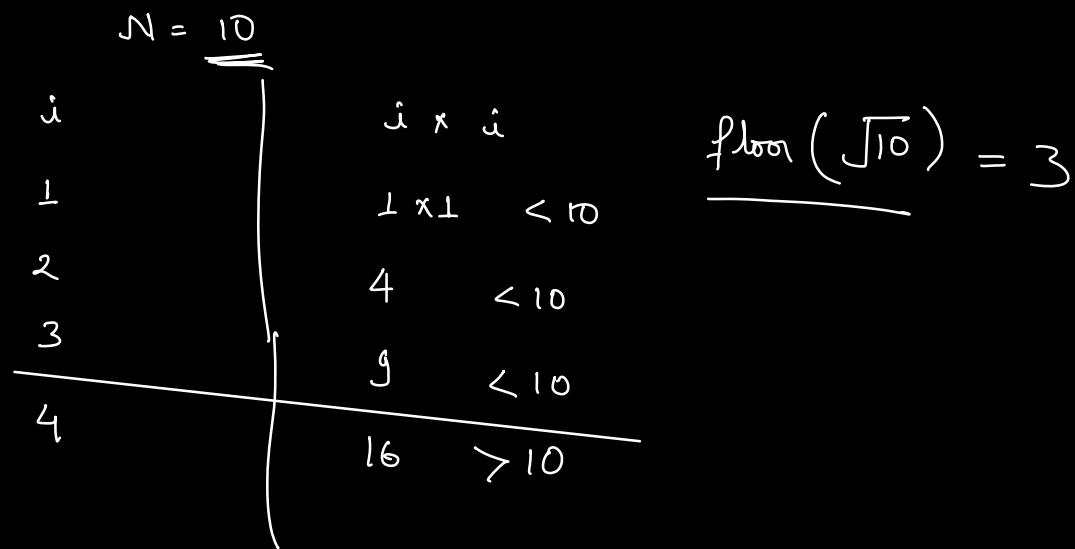
Q5

```
void fn (N) {
    s=0;
    for (i=1; i*i <= N; i++) {
        s = s+i^2;
    }
}
```

TC :  $(\sqrt{N})$

No. of iterations =  $\sqrt{N}$

$i$	$i \times i$	$i$
1	1	< 100
2	4	< 100
3	9	< 100
;	;	;
9	81	$\geq 100$
10	100	$\in (100)$
11		100



$$i \leq \underline{\underline{\sqrt{N}}}$$

$\rightarrow \log_2 N$

$\checkmark \frac{i \times i \leq N}{\text{---}}$

$i \leq \underline{\underline{\sqrt{N}}}$

Q void fn(N) {

    int i = N;  
    a  $\rightarrow$  while ( $i > 1$ ) {  
        b  $\rightarrow$      [  $i = i/2$  ;  
        ]  
    }

No. of steps	$i(a)$	$i$ after step (b)
1	$\frac{N}{1}$	$\frac{N}{2} \rightarrow \frac{N}{2^1}$
2	$\frac{N}{2}$	$\frac{N}{4} \rightarrow \frac{N}{2^2}$
3	$\frac{N}{4}$	$\frac{N}{8} \rightarrow \frac{N}{2^3}$
4	$\frac{N}{8}$	$\frac{N}{16} \rightarrow \frac{N}{2^4}$
$\vdots$		$\vdots$
		$\textcircled{1} \leftarrow \frac{N}{2^k}$

Karun

$$\frac{N}{2^k} = 1$$

$$\Rightarrow N = 2^k$$

$$\Rightarrow \log_2 N = \log_2 2^k$$

$$\Rightarrow \boxed{\log_2 N = k} \Leftrightarrow \boxed{TC : O(\log N)}$$

$$a \times a = b$$

$$\sqrt{b} = q$$

$$\sqrt{100} = 10$$

$$10 \times 10 = 100$$

$$\sqrt{625} = 25$$

$$25 \times 25 = 625$$

$$\sqrt{64} = 8$$

$$\boxed{\log_a a^x = x}$$

$$\begin{aligned} \log_2 64 &= \log_2 2^6 \\ &= 6 \end{aligned}$$

$$\sqrt{1024} = 32$$

$$\log_2 1024 = 10$$

$$\sqrt{2^{32}} = 2^k$$

$$\log_2 2^{32} = 32$$

Q.

Void  $f_n(N) \{$

$S = 0;$

$\text{for } (i = 0; i \leq N; i = i \times 2) \{$   
 $S = S + i;$

}

}

$i$

$0 \rightarrow 0$

$i = i \times 2$

$0 = 0 \times 2$

No. of iterations =  $\infty$   $\infty$   
infinity

No. of iterations

(before)

(after)

K iterations	1	1	2 $\rightarrow 2^1$	
	2	2	4 $\rightarrow 2^2$	
	3	4	8 $\rightarrow 2^3$	
	4	8	16 $\rightarrow 2^4$	
	5	16	32 $\rightarrow 2^5$	
	$2^K$			
			$2^K \Rightarrow N$	

$$N = 2^K$$

$$\log_2 N = \log_2 2^K$$

$$\Rightarrow K = \log_2 N$$

Q      void fn(N) {

    S = 0;

    for (i = 1; i <= 10; i++) {  
        fn (j = 1; j <= N; j++) {

            S = S + i \* j;

} } }

i	j	Range
1	j	[1, N] → <u>N</u>
2	j	[1, N] → <u>N</u>
3	j	[1, N] → <u>N</u>
...	...	...
10	j	[1, N] → <u>N</u>
11		

No of iteration = 10N

TC = O(N)

Q

void fn(N) {

    S = 0;

    for (i = 1; i <= N; i++) {

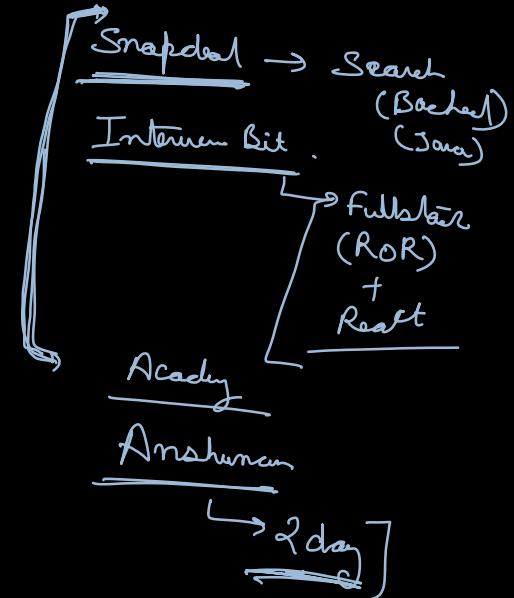
        fn (j = 1; j <= N; j++) {  
            S = S + i \* j; } } }

} }

No of iteration = N × N  
= N<sup>2</sup>

TC : O(N<sup>2</sup>)

Break tell 10:30 p



Q

void  $f_n(N)$  {

$S = 0;$

$f_n(i=0; i < N; i++)$  {

$f_n(j=0; j \leq i; j++)$  {  
 $S = S + (i \cdot j),$

}

}

i	j [0, i]	No. of iterations
0	[0, 0]	1
1	[0, 1]	2
2	[0, 2]	3
3	[0, 3]	4
⋮	⋮	⋮
N-1	[0, N-1]	N
N	---	0

$$1 + 2 + 3 + 4 + \dots + N \Rightarrow AP$$

$$\text{No of iterations} = \frac{N \times (N+1)}{2} = \frac{N^2 + N}{2} \quad TC : O(N^2)$$

Q

void fn(N) {

for (i=1; i<=N; i = i+2) {

for (j=1; j<=i; j++) {

Print(i+j);

}

}

i	j [1, i]	No. of iterations
1	[1, 1]	1
3	[1, 3]	3
5	[1, 5]	5
7	[1, 7]	7
9	[1, 9]	9
:	:	:

AP

a = 1

d = 2

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

$\underbrace{1 + 3 + 5 + 7 + 9}_{\text{Sum of first } N \text{ odd numbers}} \dots N$

$$S_n = \left( \frac{n+1}{2} \right)^2 = \frac{N^2 + 1^2 + 2N}{4} = \frac{N^2 + 2N + 1}{4}$$

$\rightarrow$  Jokes

Extra Rab (Modij)

Q

void fn(N) {

for (i=1; i<=N; i++) {

for (j=1; j<=N; j=j\*2) {

Print(i\*j);

}

}

i	j	No. of iterations
1	[1, N]	$\log_2 N$
2	[1, N]	$\log_2 N$
3	[1, N]	$\log_2 N$
:	:	:
N	[1, N]	$\log_2 N$

$1 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 16 \rightarrow 32 \rightarrow 64 \dots N$

$\underbrace{\hspace{10em}}$   
 $\log_2 N$

No. of iterations =  $\underbrace{N \cdot \log_2 N}$   
 $T.C. = O(N \log N)$

Q

void fn(N){

$S = 0;$

$fn(i=1;$

$i <= (\underbrace{1 \ll N}) ; i++) \{$   
 $S = S + i;$

$[1, \underline{2^N}]$

$\boxed{1 \ll N \Rightarrow 2^N}$

$(i < N) \quad X$

$\quad \quad \quad 2^N$

}

$for (i=1 ; i <= X ; i++) \}$

$\quad \quad \quad =$

No of iterations  $\Rightarrow 2^N$

$T.C. : O(2^N)$

$X \Rightarrow 2^N$

Q

$fn(N) \{$

$S = 0,$

$fn(i=1 ; i \leq N ; i++) \{$

$for (j=1 ; j \leq (\underbrace{1 \ll i}) ; j++) \{$   
 $S = S + i * j;$

$\quad \quad \quad \}$

$i$	$j$ $[1, 2^i]$	No of iterations
1	$[1, 2^1]$	$2^1$
2	$[1, 2^2]$	$2^2$
3	$[1, 2^3]$	$2^3$
4	$[1, 2^4]$	$2^4$
$\vdots$	$\vdots$	$\vdots$
$N$	$[1, 2^N]$	$2^N$

$$2^1 + 2^2 + 2^3 + \dots + 2^N \quad \underline{\text{GP}}$$

$$a = 2$$

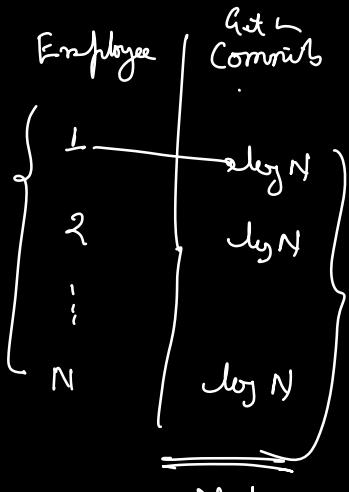
$$r = 2 \\ t = N$$

$$S_n = \frac{a(r^t - 1)}{r-1}$$

$$S_n = \frac{2(2^N - 1)}{2-1} = 2(2^N - 1)$$

No. of iterations :  $2(2^N - 1)$

TC :  $O(2^N)$



Big-O (Time Complexity) [What?]

- ① Calculate no of iterations ✓
- ② Neglect all the lower order terms (only consider the highest power)
- ③ Neglect all constants

Ex 1.  $4n^2 + 3n + 1 \Rightarrow O(n^2)$

2.  $3n\sqrt{n} + 4\log n + 31n \log n$

$$4\log n < 31n \log n < 3n\sqrt{n}$$

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

$\mathcal{O}(N/2)$  }  $\mathcal{O}(\cancel{N})$  } Google       $\text{int add}(a, b) {$   
 $c = \cancel{a \times b};$   
 $\text{int } c + q;$   
 $\cancel{\mathcal{O}(1)} \Rightarrow \cancel{\mathcal{O}(2)}$

Recursion       $\{ \text{for } (i=0 ; i < N; i+2) { }$  }  $\mathcal{O}(N)$   
 $\downarrow$   
 $\mathcal{O}(N)$

$$\begin{bmatrix} AP \\ GP \\ \log \end{bmatrix}$$