1) Time complexity & Space Complexity
2) Assymptotic analysis (Big O)
3) Big O → meaning
4) Time limit exceed (TLE)

$\Big\}$ TC II

Maths Concepts

1. $\log_2 N$ → No of times we need to divide N by 2 to reduce it to 1.

Quiz 2 [3, 10] ⇒ 3, 4, 5, 6, 7, 8, 9, 10
⇒ 8

[a, b] ⇒ b − a + 1
[1, 7] ⇒ 7 − 1 + 1 ⇒ 7
[12, 23] ⇒ 23 − 12 + 1 ⇒ 12

3. Arithmatic Progression

$\underbrace{4, \; 7}_{3}, \; \underbrace{7, \; 10}_{3}, \; \underbrace{10, \; 13}_{3}, \; \underbrace{13, \; 16}_{3} \; \ldots$

$\underbrace{a, \; a+d}_{d}, \; \underbrace{a+d, \; a+2d}_{d}, \; \underbrace{a+2d, \; a+3d}_{d}, \; \underbrace{a+3d, \; a+4d}_{d} \; \ldots$

$a \rightarrow$ first term

$d \rightarrow$ Common diff

$N \rightarrow$ no. of terms

$$\boxed{\text{Sum of first } N \text{ terms} = \frac{N}{2}\left[2a + (N-1)d\right]}$$

Sum of first N natural no. $= \dfrac{N(N+1)}{2}$

4. Geometric Progression (G.P.)

$$\overbrace{5, \; 10}^{}_{10/5}, \; \overbrace{10, \; 20}^{20/10}, \; \underbrace{20, \; 40}_{40/2}, \; \overbrace{40, \; 80}^{80/40}, \; \underbrace{80, \; 160}_{160/80} \; \ldots\ldots$$

$a, \quad ar, \quad ar^2, \quad ar^3, \quad ar^4 \; \text{-----}$

$a \rightarrow$ first term

$r \rightarrow$ Common ratio

$N \rightarrow$ no. of term.

Sum of first N terms of GP $= \dfrac{a(r^N - 1)}{r-1}$ $\boxed{r > 1}$

Log Basic

$\log_a a^x = x \quad \Big| \quad \log_2 2^{10} = 10 \quad \Big| \quad \log_{10} 10^{12} = 12$

Q int fn (N) {

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

$i : [1, N]$

No of iterations = N

TC : O(N)

Q

```
void fn (int N, int M) {
    for (i = 1; i <= N; i++) {
        if (i % 2 == 0) {
            Print (i);
        }
    }

    for (j = 1; j <= M; j++) {
        if (j % 2 == 0) {
            Print (j);
        }
    }
}
```

$\Rightarrow$ $i : [1, N]$

No of iteration = N

$\Rightarrow$ $j : [1, M]$

No. of iteration = M

Total No of iterations

= N + M

TC : O(N+M)

Q3

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

i

odd: 1 → i = i+2, 3 → i = i+2, 5 → i = i+2, 7, 9, 11, 13, ⋮

iteration = No. of odd no. in the range [1, N]

How many odd no. from 1 to N

N = 7 : {1, 2, 3, 4, 5, 6, 7} ⇒ 4

N = 6 : {1, 2, 3, 4, 5, 6} ⇒ 3

No. of odd no. from 1 to N = $\frac{(N+1)}{2}$ = $\frac{N}{2} + \frac{1}{2}$

No of iterations = $\frac{(N+1)}{2}$

TC : O(N)

Q

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

i : [0, 100]

100 − 0 + 1 ⇒ 101

No of iteration = 101

$101 \times N^0 \Rightarrow 1$

TC : O(1)

$\boxed{\text{Constant TC}}$

Q

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

$i <= \sqrt{N}$

$i * i <= N$

$i^2 <= N$

$i <= \sqrt{N}$

$i : [1, \sqrt{N}]$

No of iterations $= \sqrt{N}$ TC: $O(\sqrt{N})$

Q 6

```
void fn(N) {
    int i = N;
    while (i > 1) {
        i = i/2;
    }
}
```

i Before	Iteration	i after	i = i/2
N	1	N/2	$N/2^1$
N/2	2	N/4	$N/2^2$
N/4	3	N/8	$N/2^3$
N/8	4	N/16	$N/2^4$

⋮

$1 \rightarrow \frac{N}{2^K}$

$$i : \underbrace{N \rightarrow \overbrace{\frac{N}{2} \rightarrow \frac{N}{4} \rightarrow \frac{N}{8} \rightarrow \frac{N}{16} \text{ ------ } 1}^{\log_2 N - 1}}_{K \text{ iteration}}$$

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

$$\Rightarrow 2^K = N$$

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

$$\Rightarrow \boxed{K = \log_2 N}$$

No of iteration $= \log_2 N$

TC: $O(\log N)$

Q

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

i_b	iter	i_a
0	1	0
0	2	0
0	3	0
0	4	0
⋮	⋮	⋮

No of iterations = ∞

∞

Q

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

i_b	iter	i_a
1	1	$2 \rightarrow 2^1$
2	2	$4 \rightarrow 2^2$
4	3	$8 \rightarrow 2^3$
8	4	$16 \rightarrow 2^4$
		⋮
		$N \rightarrow 2^k$

$i = 1, 2, 3, 4, \ldots\ldots N$

$\xrightarrow{\times 2}$

$\xleftarrow{/2}$

$\log_2 N$

$2^k = N$

$\Rightarrow K = \log_2 N$

TC: $O(\log N)$

$N = 32 \xrightarrow{/2} 16 \xrightarrow{/2} 8 \xrightarrow{/2} 4 \xrightarrow{/2} 2 \xrightarrow{/2} 1$

$\times 2 \quad \times 2 \quad \times 2 \quad \times 2 \quad \times 2$

Q

```
void fn (N) {
    for ( i= 1;   i<= 10;   i++ ) {
        for (j=1;   j<=N;   j++) {
            Print (i*j);
        }
    }
}
```

i	j	Inner Loop iteration
1	[1, N]	N +
2	[1, N]	N +
3	[1, N]	N +
⋮	⋮	⋮
10	[1, N]	N +
11	×	0

No of iteration = 10N

TC : O(N)

Q

```
void fn (N) {
    for ( i= 1;   i<= N ;   i++ ) {
        for (j=1;   j<=N;   j++){
            Print (i*j);
        }
    }
}
```

i	j	Inner Loop iteration
1	[1, N]	N +
2	[1, N]	N +
3	[1, N]	N +
⋮	⋮	⋮
N	[1, N]	N
N+1	×	×

No of iteration = N^2

TC: $O(N^2)$

Q

```
void fn (N) {
    for (i = 1; i <= N; i++) {
        for (j = 1; j < N; j = j*2) {
            Print (i*j);
        }
    }
}
```

$\underline{\log_2 N}$ $\underline{j<=N}$

i	j	iterations
1	1 → N	$\log_2 N$ +
2	1 → N	$\log_2 N$ +
3	1 → N	$\log_2 N$ +
⋮	⋮	⋮
N	1 → N	$\log_2 N$
N+1	0	0

No of iterations = $N \log N$

TC : $O(N \log N)$

Q

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

$i : [1, 2^n]$

# of iterations = 2^n

TC : $O(2^n)$

Q

```
void fn (N) {
    for (i = 1; i <= N ; i++) {
        for (j = 1, j <= 2^i ; j++) {
            Print (i+j);
        }
    }
}
```

i	$j : [1, 2^i]$	Iteration	
1	[1, 2]	2 +	2^1
2	$[1, 2^2]$	4 +	2^2
3	$[1, 2^3]$	8 +	2^3
4	$[1, 2^4]$	16 +	2^4
5	$[1, 2^5]$	32	2^5
⋮	⋮	⋮ +	
N	$[1, 2^N]$	2^N	2^N

# of iterations = $\underbrace{2 + 4 + 8 + 16 + 32 + \cdots\cdots + 2^N}_{GP}$

$a = 2$

$r = 2$

No of terms = N

Sum = $a \times \left(\dfrac{r^N - 1}{r - 1}\right)$

$= \dfrac{2\,(2^N - 1)}{2 - 1}$

$= 2(2^N - 1)$ TC : $O(2^N)$

# of iterations = $2(2^N - 1)$

$\overbrace{2^1,\ 2^2,\ 2^3,\ 2^4\ \cdots\cdots\ 2^N}^{N \text{ terms}}$

↑ N^{th} term

How to calculate Big O Notation from no. of iterations

1) Neglect all lower order terms
2) Neglect all constant coefficient terms

Iterations = $4N^2 + 3N + 1$

↳ Neglect

$4N^2 \longrightarrow N^2 \Rightarrow O(N^2)$

Q $4N^2 + 3N + 10^6$

$\longrightarrow N^2$

Q $3N\sqrt{N} + 4\log N + 31\, N\log N \Rightarrow O(N\sqrt{N})$

$N = 2^{32}$

$N\sqrt{N}$	$N\log N$
$2^{32} \times 2^{16}$	$2^{32}\log_2 2^{32} \Rightarrow 2^{32} \times 2^5$
2^{48}	$= 2^{37}$

$O(1) < \log_2 N < \sqrt{N} < N < N\log N < N\sqrt{N} < N^2 < 2^N < N! < N^N$

$N = 32$

$\log_2 32$	$\sqrt{32}$	32	$32\log_2 32$	$32 \times \sqrt{32}$	32^2	2^{32}	$32!$
5	5.65	32	160	181	1024	4×10^9	$\gg 4 \times 10^9$

DOUBTS

```
void fn(N){
    int i = N;
    while (i >= 1)
        i = i/2;
    }
}
```

$N \to \frac{N}{2} \to \frac{N}{4} \to \frac{N}{8} \cdots\cdots \to 1$

$i \to i/2$

$\log_2 N$

$i >= 1$

$i_{min} \Rightarrow 1$

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

$\log_2 N$

$\underbrace{1 \to 2}_{\times 2} \to 4 \to 8 \cdots\cdots \to N$

$i <= N$

$N = 32$

$\cancel{32} \xrightarrow{\times 2} \cancel{16} \xrightarrow{\times 2} \cancel{8} \xrightarrow{\times 2} \cancel{4} \xrightarrow{\times 2} \cancel{2} \xrightarrow{\times 2} \cancel{1}$

$\times [a, b] \Rightarrow [1, 10] \Rightarrow 10$

$\frac{a}{n}$

N^a

$1 \to 2^0$

$2 \to 2^1$

$4 \to 2^2$

$8 \to 2^3$

$\vdots$

$N \to 2^K$

K^{th} term

$O(\log N)$

$\Rightarrow N = 2^K$

$\Rightarrow \boxed{\log_2 N = K}$