2 session → TC1 & TC2

* Time complexity & space complexity
* Asymptotic analysis
* Big O
* TLE

** Maths Concepts

1) $\log_2 N$:- no. of times we need to divide a no. by 2 till it reaches 1.

2) $x \in [a, b]$ → x belongs b/w a & b inclusive

$x \in [3, 7] \Rightarrow 3\ 4\ 5\ 6\ 7$

$x \in (a, b)$ → x belongs b/w a & b exclusive

$x \in (3, 7) \Rightarrow 4\ 5\ 6.$

$x \in [3, 7) \Rightarrow 3, 4, 5, 6$

$x \in (3, 7] \Rightarrow 4, 5, 6, 7.$

3) Arithmetic progression (A.P) :-

ranges of nos. which are exactly 'd' apart

⇒ 4 7 10 13 16 19 22 ⇒ A.P

3 3 3 3 3 3

⇒ 2 6 10 14 18 22 26 ⇒ A.P

4 4 4 4 4 4

⇒ 3 13 23 33 43 53 ⇒ A.P

10 10 10 10 10

⇒ 2 4 5 7 8 11 16 17 19 ⇒ ~~A.P~~

2 1 2 1 3 5 1 2

4+2*3 4+4*3

eg ⇒ 4 7 10 13 16 19 22

4+3 4+3*3

A.P ⇒ a is the first term

d is the diff.

$a = 4$

$d = 3$

⇒ $a,\ a+d,\ a+2d,\ a+3d,\ a+4d$ ___ ___ ___

$\downarrow$ 1, $\downarrow$ 2, $\downarrow$ 3, $\downarrow$ 4, $\downarrow$ 5

$\boxed{N^{th}_{element} = a + (n-1)d}$

Sum of first N terms of an AP $\Rightarrow \frac{N}{2}\left[2a + (N-1)d\right]$

Sum of first N terms natural nos. $\Rightarrow \frac{N(N+1)}{2}$

$1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \; - - - \; -$

$a = 1$

$d = 1$

$\frac{N}{2}\left[2*a + (N-1)*d\right]$

$= \frac{N}{2}\left[2 + (N-1)\right]$

$\Rightarrow \frac{N}{2}\left[N+1\right] = \frac{N(N+1)}{2}$

4) Geometric Progression (G.P)

ratio of neighbours are same.

$\Rightarrow \quad 5 \quad 10 \quad 20 \quad 40 \quad 80 \quad 160 \quad 320 \; - - - \; - \Rightarrow$ G.P

$5 \quad 5*2 \quad 5*2*2 \quad 5*2*2*2 \quad 5*2*2*2*2$

$5*2^0 \quad 5*2^1 \quad 5*2^2 \quad 5*2^3 \quad 5*2^4$

first term $= a$

common ratio $= r$

GP $\Rightarrow$ a, ar, ar^2, ar^3, ar^4 ----

$$\boxed{\text{Sum of first N terms} = \frac{a(r^N - 1)}{(r-1)}}$$

$r > 1$

5) log

$\log_2 x \Rightarrow$ dividing x by 2 till it reaches 1.

$\log_3 x \Rightarrow$ dividing x by 3 till it reaches 1.

$\log_a x \Rightarrow$ dividing x by a till it reaches 1.

$$\boxed{\log_a a^m \Rightarrow m}$$

$\log_a a^m \Rightarrow$ dividing a^m by a till it reaches 1

$\downarrow$

ex

$\log_{10} 10^{12} \Rightarrow 12$

$\frac{10^{12}}{10} = 1.$

$10 * 10 * 10$ ---- 10 (10)

Q. What is no. of iterations ? $\Rightarrow$ $\underline{\underline{N}}$

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

$i \in [1, N]$

$= N-1+1$

$= \underline{\underline{N}}$

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

$x \in [3, 7] \Rightarrow 7-3+1 = 5$

$\downarrow$

3 4 5 6 7

Q.2.

```
void fun(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)
    }
}
```

$i \in [1, N]$

$\Rightarrow N-1+1$

$\Rightarrow \underline{N}$

$j \in [1, M]$

$\Rightarrow M$

iterations = $N+M$

Q3.

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

i
1 → i+2
3 → +2
5 → +2
7 → +2
9 → +2
11
⋮
N

[1, N] → $\underline{\underline{\text{Odd nos.}}}$ = $\underline{\underline{\text{iterations}}}$

$N = 7 \Rightarrow \{1, 2, 3, 4, 5, 6, 7\} \Rightarrow 4.$

$N = 6 \Rightarrow \{1, 2, 3, 4, 5, 6\} \Rightarrow 3$

N is even $\Rightarrow$ no. of odd $\Rightarrow n/2$

N is odd $\Rightarrow$ no. of odd nos. $\Rightarrow n/2 + 1.$

Total no of odds $[1, N] \Rightarrow \frac{(N+1)}{2}$

eg $\Rightarrow N = 7 \Rightarrow \frac{(7+1)}{2} = 8/2 = 4$

$N = 6 \Rightarrow \left(\frac{6+1}{2}\right) = 7/2 = 3$

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

Q.4.

```
int fn(N) {
    s=0;
    for( i=0; i<=100; i++){
        s= i+ i2 +1
    }
    return s;
}
```

iterations :- $i \in [0, 100]$

$= 100 - 0 + 1$

$= 101$

Q5. void fun(N){

s = 0

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

s = i^2

}

return S;

}

iteration $\Rightarrow i \in [1, \sqrt{N}]$

$\Rightarrow \sqrt{N}$

$i * i <= N$

$\Rightarrow i^2 <= N$

$\Rightarrow \boxed{i <= \sqrt{N}} \rightarrow i_{max} = \sqrt{N}$

Q6. void fun(N){

int i=N;

$\Rightarrow$ while(i>1){

i = i/2;

}

}

$\Rightarrow \boxed{\text{iterations} = \log_2 N}$

i_{before}	i_{After}	iteration
N	N/2	1
N/2	N/4	2
N/4	N/8	3

1 2 3 4

$N \to N/2 \to N/4 \to N/8 \to - - - - \to 1$

$N/2^0 \quad N/2^1 \quad N/2^2 \quad N/2^3 \quad - - - \quad N/2^k$

$$1 = N/2^k$$

$$\Rightarrow 2^k = N$$

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

$$\Rightarrow k = \log_2 N$$

Q.7. void fun(N){

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

iteration	i_B	i_A
1	0	0
2	0	0
3	0	0
4	0	0

iteration ⇒ infinite

Q 8. void fun(N) { → iterations = $\log_2 N$

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

iterat	i_B	i_A
1	1	2 → 2^1
2	2	4 → 2^2
3	4	8 → 2^3
4	8	16 → 2^4

⋮

$N \to 2^k$

$$\underbrace{N \to N/2 \to N/4 \to N/8 - - - - - \to 1}_{\log N}$$

$$\underbrace{N \leftarrow N/2 \leftarrow N/4 \leftarrow - - - - - \leftarrow 8 \leftarrow 4 \leftarrow 2 \leftarrow 1}_{\log N}$$

$N = 2^k$

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

$\Rightarrow \boxed{\log N = k}$

12

Q. 9. a) void fun(N) {

for (i=1 ; i<=10 ; i++) {

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

print (i*j)

}

}

}

iteration => 10N

i	j	no. of iteration
1	[1,N]	N
2	[1,N]	N
3	[1,N]	N
⋮	⋮	N
⋮	⋮	⋮
10		N

10*N

Q b) void fun(N) {

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

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

print (i*j)

}

}

,

}

iterations => $x * y$

Q. 10. void fun(N) {

```
for ( i=1; i<=N ; i++) {
    for ( j=1; j<=N; j++) {
        print ( i*j)
    }
}
}
```

iterations => N^2

i	j	iterations
1	[1, N]	N
2	[1, N]	N
3	[1, N]	N
⋮	⋮	⋮
N	⋮	N

$N * N$

Q.11 void fun(N){

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

i	j	iteration
1	$1 \to N$	$\log_2 N$
2	$1 \to N$	$\log_2 N$
3	$1 \to N$	$\log_2 N$
⋮		⋮
N		$\log_2 N$

iteration $\Rightarrow N \log N$

Q.12 void fun(N){

```
for( i=1; i<=2^N; i++){
    print(i)
}
}
```

iterations $\Rightarrow i \in [1, 2^N] \Rightarrow 2^N - 1 + 1$

$\Rightarrow 2^N$

Q.13. void fun(N) {

```
for ( i=1 ; i<=N ; i++) {
    for ( j=1 ; j<=2^i ; j++ ) {
        print(i+j)
    }
}
}
```

$[1, 2^i]$

i	j	iterations
1	$[1, 2^1]$	2
2	$[1, 2^2]$	4
3	$[1, 2^3]$	8
4	$[1, 2^4]$	16
⋮		
N	$[1, 2^N]$	2^N

$$2 + 4 + 8 + 16 + \dots + 2^N$$

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

$a = 2$

$r = 2$

$$\longrightarrow \frac{2(2^N - 1)}{(2-1)} \Rightarrow \underline{\underline{2(2^N - 1)}}$$

iterations

$$\frac{a(r^N - 1)}{(r-1)}$$ { Sum of N terms of GP }

U ,

** How to calculate Big O notation from no. of iterations

no. of iterations ⇒ fn(N) ← input

Step 1) Neglect all lower order terms

Step 2) Neglect all constant terms.

ex ⇒ iterations = $4N^2 + 3N + 1$

$\Rightarrow 4N^2$

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

ex ⇒ $10N^2 + 3\log N$

$\Rightarrow 10N^2$

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

ex ⇒ $3 * 2^N + 10N$

$\Rightarrow 3 * 2^N$

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

eq $\Rightarrow$ $3N\sqrt{N} + 4\log N + 31 N\log N$

$N = 2^{32}$ [take large values]

$N\sqrt{N} \Rightarrow 2^{32} * \sqrt{2^{32}} = 2^{32} * 2^{16} = 2^{48}$

$N\log N \Rightarrow 2^{32} * \log_2 2^{32} = 32 * 2^{32}$

$= 2^5 * 2^{32}$

$= 2^{37}$

$3N\sqrt{N} + 4\log N + 31 N\log N$

$\Rightarrow 3N\sqrt{N}$

$\Rightarrow N\sqrt{N} \Rightarrow N^{3/2} \Rightarrow O(N\sqrt{N})$

or $O(N^{3/2})$

* <u>Comparing time complexities:</u>

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

$< O(N^2) < O(2^N) < O(N!)$

$< O(N^N)$

$N = 32$

$O(1) = \text{const}$

$O(\log_2 N) = \log_2 32 = 5$

$O(\sqrt{N}) = \sqrt{32} = 5.65\ldots$

$O(N) = 32$

$O(N \log N) = 32 * \log_2 32 = 32 * 5 = 160.$

$O(N^2) = (32)^2 = 1024$

$O(2^N) = 2^{32}$

$O(N!) = 32!$

$O(N^N) = 32^{32}$