Agenda:-

* Time complexity & space complexity
* Asymptotic analysis (Big O)
* Big O → meaning
* TLE (Time Limit Exceeded).

Maths Concepts :-

1) $\log_2 N$:- no. of times we need to divide N by 2 to get 1.

2) $x \in [a, b]$ → inclusive ⇒ $(b-a+1)$ $[b>a]$.

eg ⇒ $x \in [3,6]$ ⇒ x ⇒ 3, 4, 5, 6.

$x \in (a, b)$ → exclusive

eg ⇒ $x \in (3,6)$ ⇒ x → 4, 5.

3) Arithmetic Progression (A.P) :-

in a series, each number is 'd' apart.

4 7 10 13 16 19 22

3 3 3 3 3 3

4, 4+3, 4+2*3, 4+3*3, 4+4*3, 4+5*3.

$a, a+d, a+2d, a+3d, a+4d$ ——— first term = a

diff. = d.

Sum of first N terms = $N/2\,[2a + (N-1)d]$

eg ⇒ 11 13 15 17 19 21 - —— -

$a = 11$

$d = 2$

$N = 10$

Sum of first 10 terms.

$$\frac{10}{2}(2*11 + (10-1)2)$$

$$= 5*(22+18)$$

$$= 5 * 40$$

$$= 200$$

1 2 3 4 5 6 7 8 9 - — - N.

$a = 1$

$d = 1$

N.

$$N/2\,[2a + (N-1)d]$$

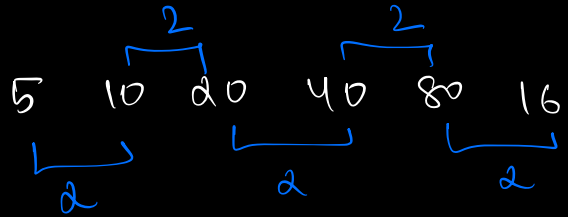
$$N/2\,[2(1) + (N-1)(1)]$$

$$= N/2\,(2 + N - 1)$$

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

4. Geometric Progression :- (G.P)

In a series, ratio of every two nos. are same.
(neighbours)



↓

$5, 5\times2, 5\times2^2, 5\times2^3, 5\times2^4, \ldots$ first term = a, ratio = r.

$a, ar, ar^2, ar^3, ar^4 \ldots$

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

5 log :-

$\log_2^N$ ⇒ dividing N by 2 kb tk it reaches 1.

$\log_3^N$ ⇒ dividing N by 3. kb tk it reaches 1.

$\log_x^N$ ⇒ dividing N by x. kb tk it reaches 1.

Q1. No. of iterations :-

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

$i \in [1, N]$

$= N-1+1$

$= N$

N iterations

$O(N)$

Q2. No. of iterations :-

```
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)
    }
}
```

$i \in [1, N]$

$= N-1+1$

$= N$ iterations

$j \in [1, M]$

$= M-1+1$

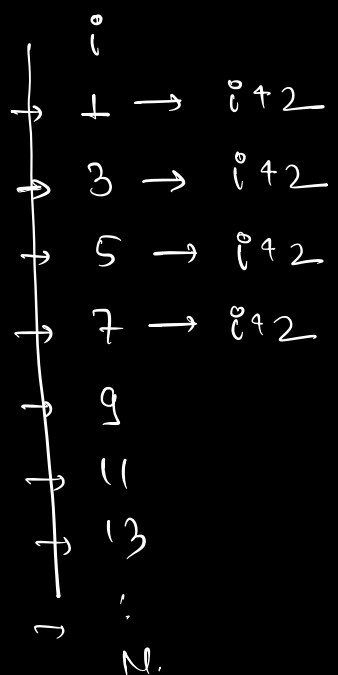
$= M$ iterations

total iterations $= N+M$

$O(\max(N, M))$

Q3.

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



no. of iterations for $i \in [1,13]$ = 7
no. of odd nos. from [1,13] = 7

No. of iterations = no. of odd nos. in range [1,N].

if, N is even, no. of odd nos. => $N/2$

N is odd, no. of odd nos. => $N/2 + 1$.

total no. of odds in $[1,N] = \left(\frac{N+1}{2}\right)$ O(N)

Ex => N = 8. = $\left(\frac{8+1}{2}\right) = 9/2 = 4$

N = 11 => $\left(\frac{11+1}{2}\right) = \frac{12}{2} = 6$

~~1~~ 2 ~~3~~ 4 ~~5~~ 6 ~~7~~ 8 ~~9~~ 10 ~~11~~

iteration	i
1	1
2	3
3	5
4	7

Q4.

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

$i \in [0, 100]$

$= 100 - 0 + 1$

$= 101$

↑ iteration

$O(1)$

Q5.

```
void fun(N) {
    s = 0;
    for (i = 1; i * i <= N; i++) {
        s = s + i2
    }
    return s;
}
```

$i * i <= N$

$\Rightarrow i^2 <= N$

$\Rightarrow i <= \sqrt{N}$

$i_{max} \Rightarrow \sqrt{N}$

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

$= \sqrt{N} - 1 + 1 = \sqrt{N}$ iterations

$O(\sqrt{N})$

Q 6.

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

i Before	i After	no. of iter
N	N/2	1
N/2	N/4	2
N/4	N/8	3
N/8	N/16	4.
	⋮	
	1	

iterations = $\log_2 N$

$O(\log N)$

Q 7.

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

infinite iterations

i Befor	i After	iter
0	0	1
0	0	2
0	0	3
(	?	?

∞

Q8.

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

i_B	i_A	No. of itr.
1	$2 \to 2^1$	1
2	$4 \to 2^2$	2
4	$8 \to 2^3$	3
8	$16 \to 2^4$	4
⋮	⋮	⋮
	$N \to 2^k$	k

$$2^k = N$$

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

$$\Rightarrow k = \log_2 N \quad O(\log_2 N)$$

$\downarrow$

no. of iterations

Q9.

```
void fun(N) {
    x ⇒ for(i = 1; i <= 10; i++) {
        y ⇒ for(j = 1; j <= N; j++) {
            print(i * j)
        }
    }
}
```

i	j	iterat
1	[1, N]	N
2	[1, N]	N
3	[1, N]	N
⋮	⋮	⋮
10	[1, N]	N

10 N

O(N) iterations = 10 N total => x x y

Q 10. void func (N) {

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

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

print (i*j)

}

}

}

total iterations = N * N = N^2

i	j	iterations
1	[1,N]	N
2	[1,N]	N
3	[1,N]	N
:	:	:
N	[1,N]	N
		N * N

O(N^2)

Q 11.

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

iterations $\Rightarrow$ $N \log N$ $\quad O(N \log N)$

Q 12.

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

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

$= 2^N - 1 + 1$

iterations $= 2^N$ $\quad O(2^N)$

Q 13.

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

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

total Iteration ⇒ 2 + 4 + 8 + 16 + ---- + 2^N

a = 2

r = 2

N

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

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

total iterations = $2(2^N - 1)$ $O(2^N)$

#4 How to Calculate Big O notation from no. of iterations :-

⇒ function of N → iterations

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

ex ⇒ iterations = $4N^2 + 3N + 1 \Rightarrow$ TC :- $O(N^2)$

$= 4N^2$

$= N^2$

ex ⇒ iterations = $4N^2 + 3N + 10^6 \Rightarrow$ TC :- $O(N^2)$

$= 4N^2$

$= N^2$

ex ⇒ iterations = $3N\sqrt{N} + 4 \log N + 31 N \log N$.

take very large value of N.

assume, $N = 2^{32}$

$N\sqrt{N} = 2^{32} * \sqrt{2^{32}} = 2^{32} * 2^{16} = 2^{48}$. ⇐ highest order.

$N \log N = 2^{32} * \log 2^{32} = 2^{32} * 32 = 2^{32} * 2^5 = 2^{37}$

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

iterations $= 3N\sqrt{N} + 4\log N + 31N\log N \rightarrow$ tc $O(N\sqrt{N})$

$= 3N\sqrt{N}$

$= N\sqrt{N}$

** comparing time complexities [increasing order]

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

↓ very rare

$N = 32$

$O(1) \rightarrow$ const.

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

$\sqrt{N} \rightarrow \sqrt{32} \rightarrow 5.\ldots$

$N \rightarrow 32$

$N\log N \rightarrow 32 * \log_2 32 \Rightarrow 160$

$N^2 \rightarrow (32)^2 = 1024$

$2^N \rightarrow 2^{32}$

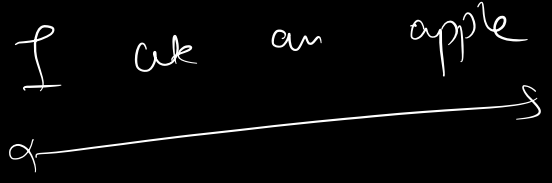
$N! \rightarrow 32!$

$N\sqrt{N} = 32\sqrt{32}$

$= 32 * (5.65)$

$\Rightarrow 180.8$

$N^N = 32^{32}$


I ate an apple