Today's Quote ⇒

The man who asks a question is a fool for a minute, the man who does not ask is a fool for life.

TC-1 → # How to calculate no. of iterations.

- → Time and Space Complexity
- → Assymptotic Analysis
- → Big-O Notation
- → TLE → Time Limit Exceeded

] TC→2.

Quiz1 :

Sum of first N natural no's →

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

Quiz 2:

$$N \to N/2 \to N/4 \to \frac{N}{8} \to \frac{N}{16} \to N/32 \;——\; 1. \to \boxed{\log_2 N}$$

$$1024 \xrightarrow[/2]{} 512 \xrightarrow[/2]{} 256 \xrightarrow[/2]{} 128 \xrightarrow[/2]{} 64 \xrightarrow[/2]{} 32 \xrightarrow[/2]{} 16 \xrightarrow[/2]{} 8 \xrightarrow[/2]{} 4$$

$$1 \xleftarrow[/2]{} 2 \xleftarrow[/2]{} 4$$

Quiz 3.:

How many no's are there in this range [3, 10] ?

$[3, 10] \rightarrow 3, 4, 5, 6, 7, 8, 9, 10$ $\boxed{\text{ans} \rightarrow 8}$

$[\rightarrow$ inclusive $\quad [a, b] = b - a + 1$

$(\rightarrow$ exclusive $\quad [a, b) = b - a$

$(a, b) = b - a - 1$.

Arithmetic Progression [A.P]

$\Rightarrow$ difference b/w any two consecutive terms is fixed. (same)

$4, 7, 10, 13, 16, 19, \ldots$

3 3 3 3 3

$a \quad a+d \quad a+2d \quad a+3d \quad a+4d \quad \ldots \quad a+(n-1)d$

first term = a

Common diff = d

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

Geometric Progression (G.P)

3   6   12   24   48   96   - -

$$3 \xrightarrow{2} 6 \xrightarrow{2} 12 \xrightarrow{2} 24 \xrightarrow{2} 48 \xrightarrow{2} 96$$

$a \quad ar \quad ar^2 \quad ar^3 \quad ar^5 \quad \text{------} \quad ar^{n-1}$

Sum of N terms of G.P $= \dfrac{a[r^n - 1]}{[r-1]}$ $r > 1.$

$\Downarrow$

$\dfrac{a[1-r^n]}{[1-r]}$ $r < 1$

first terms $\rightarrow a$

common ratio $\rightarrow r$

$$\frac{a(r^n - 1) \times -1}{(r-1) \times -1} = \frac{a(1-r^n)}{(1-r)}$$

Q1.)

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

i : 1, 2, 3, 4, — — N

: [1, N]

no. of iterations = n

Q2.)

```
void func( int N, int M){
    s=0;
    for( i = 1 ; i <= N ; i++){
        print (i);
    }
    for( j=1 ; j <= M ; j++){
        print (j)
    }
}
```

i : [1,N] → N

j : [1,M] → M

no. of iterations = N+M .

Q3.)

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

N=12.

iterations

$i = \underbrace{1 \quad 3 \quad 5 \quad 7 \quad 9 \quad 11}$ 6 $\left\lceil \frac{12}{2} \right\rceil$

$\frac{n+1}{2}$

N=13

$i = 1 \quad 3 \quad 5 \quad 7 \quad 9 \quad 11 \quad 13$ 7 $\left\lceil \frac{13+1}{2} \right\rceil$

no of iterations $= \frac{n+1}{2}$.

Q4.)

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

i : [0, 100]

$100 - 0 + 1 = 101$

no. of iterations = 101

Q5.) void fun(N){ $i <= \sqrt{N}$

$\Uparrow$

for(i=1 ; i*i <= N ; i++){ $i : [1, \sqrt{n}]$

s = 1+i2 ;

}

return s ;

}

\# no. of iterations $= \sqrt{n}$.

Q6.) void fun(int N){

i = N;

while (i > 1){

i = i / 2;

}

}

iterations	value of i
1	$N/2$
2	$N/4$
3	$N/8$
4	$N/16$
5	$N/32$
⋮	⋮

$i = N \rightarrow N/2 \rightarrow N/4 \rightarrow \frac{N}{8} \rightarrow N/16 \ldots$

$\frac{N}{2^1} \quad \frac{N}{2^2} \quad \frac{N}{2^3} \quad \frac{N}{2^4}$

After K -iterations, $\Rightarrow \frac{N}{2^K} = 1$

$\Rightarrow \boxed{N = 2^K} \Rightarrow \log_2 N = \log_2 2^K$

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

$\boxed{\#\ \text{no. of iterations} = \log_2 N}$

Q7.)

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

iteration	value of i
1	0
2	0
3	0

no. of iterations → infinite.

Q8.)

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

iterations	value of i	
1	2	$=2^1$
2	4	$=2^2$
3	8	$=2^3$
4	16	$=2^4$
5	32	$=2^5$
6	64	$=2^6$
⋮	⋮	

$i = \underbrace{2^1, \overset{2^2}{4}, 8, 16, 32, 64, ---}$

After $\underline{K}$ iterations, loop breaks.

$$i = \boxed{2^k = N}$$

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

no. of iterations $= \log_2 N$.

Q8) void fun(N){

```
void fun(N){
    s = 0;
    for( i = 1; i < N; i = i*3 ){
        s = s + i;
    }
}
```

$i = \underbrace{3^1, 3^2, 3^3, 3^4, \ldots}_{\text{After K-iterations, loop breaks.}}$

$i = [3^K = N]$

$\log_3 3^K = \log_3 N$

$$\boxed{K = \log_3 N}$$

no. of iterations $= \log_3 N$.

Nested loops

Q9.)

```
void fun (int N){
    for( i = 1; i <=10 ; i++){
        for( j=1 ; j <=N ; j++){
            print(--)
        }
    }
}
```

table.

i	j	iterations.
1	[1, N]	N ✓
2	[1, N]	+ N ✓
3	[1, N]	+ N ✓
4	[1, N]	+ N ✓
⋮		+ ⋮
10	[1, N]	+ N ✓

total no. of iterations = 10 N

Q10.)

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

table.

i	j	iterations
1	[1-N]	N
2	[1-N]	+ N
3	[1-N]	+ N
4	[1-N]	+ N
⋮		+ ⋮
N	[1-N]	+ N

$$\boxed{\begin{array}{l} 0 \text{ to } < N \\ 1 \text{ to } <= N \end{array}}$$

{same.}

total no. of iterations = N * N

Q11)

```
void fun( int N){
    for( i=0 ; i < N ; i++){
        for( j=0 ; j <= i ; j++){
            print(i*j);
        }
    }
}
```

table.

i	j	iterations
0	[0,0]	1
1	[0-1]	+ 2
2	[0-2]	+ 3
3	[0-3]	+ 4
|		+ .....
N-1	[0-N-1]	+ N

no. of iterations = $\frac{n(n+1)}{2}$

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

Q12.)

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

table.

i	j	iterations
1	[1 → N]	$\log_2 N$
2	[1 → N]	+ $\log_2 N$
3	[1 → N]	+ $\log_2 N$
|	|	
N	[1 → N]	+ $\log_2 N$

total no of iterations = $N \log_2 N$.

Q13.)

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

$i \rightarrow [1, 2, 3, 4, 5, ---\ 2^N]$

$: [1, 2^N]$

\# total no. of iterations $= 2^n$.

Q14.)

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

\#table

i	j	iterations.
1	$[1, 2^1]$	2
2	$[1, 2^2]$	$+ 2^2$
3	$[1, 2^3]$	$+ 2^3$
4	$[1, 2^4]$	$+ 2^4$
⋮		+ ⋮
N	$[1, 2^N]$	$+ 2^N$

total no of iterations $= 2^1 + 2^2 + 2^3 + ---\ 2^N$

$$= \frac{2[2^N - 1]}{[2-1]} = 2[2^N - 1]$$

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

Q.)

```
for( i=N ; i>0 ; i=i/2){
    for( j=1 ; j<=i ; j++){
        print(i*j)
    }
}
```

table.

i	j	iterations
N	[1, N]	N
N/2	[1, N/2]	N/2
N/4	[1, N/4]	N/4
N/8	[1, N/8]	N/8
⋮	⋮	⋮
1	[1, 1]	1

$\Rightarrow N = N$

$$\boxed{2^{\log_2 N} = N} \leftarrow \{\text{Assumption}\}$$

$$\Rightarrow \boxed{1 = \frac{N}{2^{\log_2 N}}}$$

total no. of iterations $= N + \frac{N}{2} + N/4 + \frac{N}{8} + \text{---} \frac{N}{2^{\log_2 N}}$

$$= N + N\left[\underbrace{\frac{1}{2^1} + \frac{1}{2^2} + \frac{1}{2^3} + \text{----} \frac{1}{2^{\log_2 N}}}\right].$$

first term[a] : $1/2$

no. of terms : $\log_2 N$

common ratio : $1/2$

$$\text{Sum} = \frac{a[1 - r^n]}{[1 - r]}$$

$$= \frac{\cancel{\frac{1}{2}}\left[1 - \left(\frac{1}{2}\right)^{\log_2 N}\right]}{\cancel{1/2}}$$

$$= 1 - \frac{1}{2^{\log_2 N}} = 1 - \frac{1}{N} = \frac{N-1}{N}.$$

no. of iterations $= N + \cancel{N}\left[\frac{N-1}{\cancel{N}}\right] = \boxed{2N-1}$

<u>for very large value n.,</u>

compare $\sqrt{N}$ and N $\Rightarrow \sqrt{N} < N$

compare $\sqrt{n}$ and $\log_2 N$ $\Rightarrow \log_2 N < \sqrt{N} < N$

$$\{ \log_2 N < \sqrt{N} < N < N \log_2 N < N\sqrt{N} < N^2 < 2^N \}.$$

<u>How to write Big-O notation?</u>

what is Big-O ?
why Big-O ? $\Rightarrow$ next class.

1. Calculate no. of iterations.
2. Neglect lower order terms.
3. Neglect constant / co-efficients.

$$f(N) = 10N^2 + \underbrace{100N^1}_{\alpha} + \underbrace{10^3 \cdot N^0}_{\alpha}$$

$$= 10N^2$$

$$\Rightarrow \boxed{O(N^2)}$$

$$\cancel{4}N^2 + \cancel{3N + 10^6} \to O(N^2)$$

$\underbrace{4N + 3N\log N + 10^6}$

Q1 $\underbrace{4N\log N}_{} + \cancel{3}\,\underbrace{N\sqrt{N}}_{} + \cancel{10^6}$

$O(N\sqrt{N})$

A.P, G.P.

{ 10^{th} mathematics }