Today's Content

T.C → 2
- → Big O - O ( )
- → TLE - (Time limit Exceeded error)
- → Why TLE ?

TC → 1 { How to calculate iterations → 15/17 questions

← Quote →

Don't tell your god how big your storm is.
Tell your storm how big your god is.

Basic maths Revision

Quiz 1: Sum of Natural numbers:

$S_N = 1 + 2 + 3 + \dots N = \frac{n(n+1)}{2}$.

→ : $[3\ 10] \rightarrow \{3, 4, 5, 6, 7, 8, 9, 10\} \rightarrow 8$

$[4\ 8] = \{4, 5, 6, 7, 8\} \rightarrow 5$

No. of elements from $[a\ b]$ included

: $[a\ b] = b - a + 1$ #imp.

GP (when we divide any 2 cons. elements, ratio have to be same)

↳ Geometric Progression

$S_1 =$ 3 6 12 24 48 96

ratio = $\frac{6}{3} = \frac{12}{6} = \frac{24}{12} = \frac{48}{24} = \frac{96}{48} = 2$

$S_2 =$ 2 6 18 54 162

ratio = $\frac{6}{2} = \frac{18}{6} = \frac{54}{18} = \frac{162}{54} = 3$

// given GP

1st	2nd	3rd	...	Nth
a	ar	ar^2		ar^{n-1}

first term = a

common ratio = r

$$S_n = a + ar + ar^2 + \cdots ar^{n-3} + ar^{n-2} + ar^{n-1}$$

→ multiply both sides with r :-

$$S_n r = ar + ar^2 + ar^3 + \cdots ar^{n-2} + ar^{n-1} + ar^n$$

$$-\left(S_n = a + ar + ar^2 + \cdots ar^{n-3} + ar^{n-2} + ar^{n-1}\right)$$

$$rS_n - S_n = ar^n - a$$

$$\Rightarrow S_n(r-1) = a(r^n-1)$$

$$\Rightarrow \boxed{S_n = \frac{a(r^n-1)}{r-1}}$$

Ex:- $a = 5,\ r = 2,\ N = 5$

→ 5, 10, 20, 40, 80 => 155

Lets use formula

$$\frac{a(r^n-1)}{r-1} \Rightarrow \frac{5(2^5-1)}{2-1} \Rightarrow 5 * 31$$

$$\Rightarrow 155$$

Q1 Quiz

```
void func (int N) {
    int S = 0;
    (i=0; i<=100; i++) {
        S = S+i
    }
}
```

i [0, 100] → 100-0+1 → 101

→ O(1)

Q2)

```
void func (int N) {
    int S = 0;
    (i=35; i<=87; i++) {
        S = S+i
    }
}
```

i [35 : 87] ⇒ 87-35+1 ⇒ 53

O(1).

Q3: Quiz

```
void func (int N) {
    int S = 0;
    (i=1; i<=N; i++) {
        S = S+i
    }
}
```

i [1 : N] → N-1+1 → N

O(N)

Q4:

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

3 → i[1 : N] → N-x+x = N

3 i[1 : M] → M-x+1 = M

→ N+M

a-b → b-a+1

06) Quiz

```
void func (int n) {
    int s = 0;
    (i=1; i*i<=n; i++) {
        s = s+i
    }
}
```

$i^2 <= n \Rightarrow i <= \sqrt{n}$

$i \in [1 : \sqrt{n}]$ $\sqrt{n}$ - x + x

$\sqrt{n}$ iter

Q7)

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

3

3

iteration	i value after iteration
1	$N/2$: $N/2^1$
2	$N/4$: $N/2^2$
3	$N/8$: $N/2^3$
4	$N/16$: $N/2^4$
k iterations	$\frac{N}{2^k}$

$i = N$

N	i
10	10 → 5 → 2 → 1 {Break}
19	19 → 9 → 4 → 2 → 1 {Break}

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

$\log a^b = b \log a$

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

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

$$\Rightarrow \log_2 N = k$$

// After $\log_2 N$ iterations code breaks.

Ques)

```
void func (int N) {
    int i = N
    while (i>0) {
        i = i/2
    }
}
```

→ Todo & Try it out.

→ No: of iterations.

Q8)

Quiz

```
void func (int n) {
    int s = 0;
    (i = 0; i <= n; i = i*2) {
        s = s + i
    }
}
```

i = 0

iterations	value of i
1	0
2	0
3	0
	.
	.
	.
	∞

infinite

Quiz

i = 1

90) → { Near approximations }

```
void func (int N) {
    int S = 0;
    (i = 1; i <= N; i = i*2) {
        S = S + i
    }
}
```

iterations	value after each iteration
1	$i = 2 = 2^1$
2	$i = 4 = 2^2$
3	$i = 8 = 2^3$
4	$i = 16 = 2^4$
⋮	
after k iterations	$i = 2^k$

$i > N$ it'll break.

$2^k > N$ after k iterations.

$\log_2 2^k > \log_2 N \Rightarrow k \log_2 2 > \log_2 N$ ($\log_2 2 = 1$)

$\Rightarrow k > \log_2 N$

Break

100)

```
void func (int N) {
    for (i = 1 ; i <= 4; i = i+1) {
        for J = 1; J <= 3; J++) {
            Print (" Hello World");
        }
    }
}
```

i	J	Total iter.
1	[1-3]	3
2	[1-3]	3
3	[1-3]	3
4	[1-3]	3
		12

iterations.

110) Quiz

```
void func (int N) {
    for (i = 1 ; i <= 10; i = i+1) {
        for (J = 1; J <= N; J++) {
            Print (" Hello World");
        }
    }
}
```

i	J	Total iter
1	[1 N]	N
2	[1 N]	N
3	[1 N]	N
⋮		⋮
10	[1 N]	N
		10N

$O(N)$

10:08:10:18 pm.

12)

Quiz

```
void func (int N) {
    for (i = 1 ; i <= N ; i = i+1) {
        for (j = 1; j <= N ; j++) {
            Print (" Hello World ");
        }
    }
}
```

i	j	Total iter
1	[1 N]	N
2	[1 N]	N
3	[1 N]	N
⋮		
N	[1 N]	N
N		N*N

iterations.

13) Quiz

```
void func (int N) {
    for (i = 1 ; i <= N ; i = i+1) {
        for (j = 1; j <= i ; j++) {
            Print (" Hello World ");
        }
    }
}
```

i	j	Total iter
1	[1 - 1]	1
2	[1 - 2]	2
3	[1 - 3]	3
⋮		+
N	[1 - N]	N
		n*(n+1)/2

iterations → $\frac{n^2}{2} + \frac{n}{2}$ → $O(N^2)$

14)

Ques 3

```
void func (int N) {
    for (i = 1; i <= N; i = i+1) {
        for (J = 1; J <= N; J = J*2) {
            Print ("Hello World");
        }
    }
}
```

i	J	Total iter
1	[1 to N]	log N
2		log N
.		
.		
N		log N
		N * log N

015)

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

i [1 : 2^N] → 2^{n-x+x}

→ 2^n iteration

Q 16)

```
void func (int N) {
    for (i = 1; i <= N; i = i+1) {
        for (j = 1; j <= 2^i; j = j+1) {
            Print ("Hello World");
        }
    }
}
```

i	j	Total iter
1	[1 to 2^1]	2^1
2	[1 to 2^2]	2^2
3	[1 to 2^3]	2^3
⋮		
N	[1 to 2^N]	2^N

$2 + 2^2 + 2^3 + \cdots 2^N$

$\sum$ $r = 2,\ a = 2,\ n = n.$

$\frac{a(r^n - 1)}{r - 1} \rightarrow \frac{2(2^n - 1)}{2 - 1} \rightarrow 2(2^n - 1)$ iterations.

Comparison functions : {large n values}

$\log n < \sqrt{n} < n < n \log n < n\sqrt{n} < n^2 < n^3 < 2^n$

$\sqrt{n} = n^{1/2}$

$n\sqrt{n} = n * n^{1/2}$

Big O → what ?

→ why ?

→ where do we use it ?

→ How do we calculate Big O for any, code

→ calculate iterations

→ Only take higher order term

→ Neglect constant coefficients.

Ex 1 :- iterations → $3N^2 + 5N + 10^4$ → $O(N^2)$

Ex 2 : iterations → $5N^2 + 10N^3 + 6N \log N + 10^6$ ⇒ $O(N^3)$

Ex 3 : iterations → $4N^2 + 3N + 10^6$ → $O(N^2)$

Ex 4 : iterations → $4N + 3N \log N + 10^6$ → $O(N \log N)$

Ex 5 : iterations → 10^3 → $O(1)$ → constant.