Time Complexity - lec1

Today Content}

- ↳ Time Complexity & Space Complexity
- ↳ Asymptotic analysis
- ↳ Big O
- ↳ TLE → Time Limit Exceeded

2 Classes

Today : How to calculate No of iterations

Q.1 # Sum of N natural no. $= \frac{N(N+1)}{2}$

Q.2 : [3, 10] ← for a given range calc no. of values in b/w

[→ inclusive

(→ exclusive

[a, b]	[a, b)	(a, b)
$b-a+1$	$b-a$	$b-a-1$
a, a+1 … , b	a, a+1, a+2 … , b-1	a+1, a+2 … b-1

Q.3 $N \rightarrow N/2 \rightarrow N/4 \rightarrow N/8 \ldots\ldots 1 \rightarrow 0$: $\boxed{\log_2 N}$

logarithm

A.P $\Rightarrow$ Arithmetic Progression. $\rightarrow$ Difference is same

Series :

4, 7, 10, 13, 16, 19, ...

$4 \xrightarrow{3} 7 \xrightarrow{3} 10 \xrightarrow{3} 13 \xrightarrow{3} 16$

In general :

1st term, 2nd, 3rd, 4th, Nth term

$a \ , \ a+d \ , \ a+2d \ , \ a+3d, \ ..... \ a+(n-1)d$

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

1st term $= a$

Common diff $= d$

$$\boxed{\log_a a^x = x}$$

$\frac{a^x}{a} \Rightarrow 1$

G.P $\Rightarrow$ Geometric progression

$\frac{6}{3} = \frac{12}{6} = \frac{24}{12} = \frac{48}{24}$

Series : 3, 6, 12, 24, 48, ...

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

$\boxed{r=2} \Rightarrow$

1st term, 2nd, 3rd, 4th, 5th, Nth term

$a, \ ar, \ ar^2, \ ar^3, \ ar^4 \ ...... \ a(r^{N-1})$

Sum of N terms in G.P. $= \frac{a \cdot (r^n - 1)}{(r-1)}$ if $(r \ != 1)$.

'r' is common ratio

Q.1

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

$\Rightarrow i = [1 \rightarrow N]$

N iterations.

$\Rightarrow O(N)$.

Q.2.

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

$\Rightarrow N$

$\Rightarrow M$

$\Rightarrow$ Total $N+M$ iterations.

$\Rightarrow O(N+M)$.

Q.3

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

$i = 1 \downarrow 3 \downarrow 5 \downarrow 7$

$i : [1, N]$

1, 3, 5, 7, 9, ...

odd numbers

$N = 10$

$i = 1, 3, 5, 7, 9 \Rightarrow 5 \Rightarrow \frac{(10+1)}{2} \Rightarrow \frac{11}{2} \Rightarrow 5$

$N = 13$

$i = 1, 3, 5, 7, 9, 11, 13 \Rightarrow 7 \Rightarrow \frac{13+1}{2}$

Iteratⁿ = No. of odd numbers from [1, N] $= \frac{(N+1)}{2}$

$O(N)$.

Q.4.

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

$i : [0, 100]$

$: 100 - 0 + 1 \Rightarrow 101$

$\Rightarrow \boxed{101\, N^0}$

⇒ Constant iteration

Independent of input (N)

$\Rightarrow \boxed{O(1)}$.

Q.5.

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

$i^2 \le N$

$i \le \sqrt{N}$

$i \to [1, \sqrt{N}]$ $\Rightarrow O(\sqrt{N})$

$\sqrt{N} - 1 + 1 \Rightarrow \sqrt{N}$

Q.6

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

N

iterations	i (value after)
1	$N/2 \Rightarrow N/2^1$
2	$N/4 \Rightarrow N/2^2$
3	$N/8 \Rightarrow N/2^3$
4	$N/2^4$
? ⇒ K^{th}	$\to N/2^K$

$\uparrow_{=1}$

$i : N \rightarrow N/2 \rightarrow N/4 \rightarrow N/8 \rightarrow \dots \boxed{1}$

when $i == 1$; loop breaks.

// After K iterations, loop breaks.

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

$\Rightarrow$ Apply $\log_2$ on both sides.

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

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

$O(\log N)$.

that means After $\log_2 N$ steps, the value of $i = 1$

After $\log_2 N$ i^{th} my value of i will become 1

Q.7

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

$i = 0 * 2 = 0$

$i \Rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0$ ... infinite.

Q.8

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

Initially $i=1$

$i=1$

iteration	i value after
1	2 2^1
2	4 2^2
3	8 2^3
4	16 2^4
kth	2^K

≥ 1

After K iterations, loop breaks.

$2^K = N$

$\log_2 2^K = \log_2 N$

$\boxed{K = \log_2 N}$ $\Rightarrow O(\log_2 N)$

$i = N$

$i > 1$

$N \to N/2 \quad N/.$

$(i=1; i<N; i = i*3)$

$i = 1$

$1 \to 3$

$2 \to 9$

$3 \to 27$

$K^{th} \to 3^K$

end

$3^K = N$

$\log_3 3^K = \log_3 N$

$\boxed{K = \log_3 N}$ $\to O(\log_2 N)$

$\log_3 N = O(\log_2 N)$

$\cancel{O(4N)}$

Nested loops

Q.9

```
void fun(N)
{
  for(i=1; i≤10; i++)
  {
     for(j=1; j≤N; j++)
     {
        print(..)
     }
  }
}
```

i	j	Iterations
1	[1 → N]	N
2	[1 → N]	N
3	[1 → N]	N
⋮	⋮	⋮
10	[1 → N]	N

Total $(10*N)$

$O(N)$

Q.10 void fun(N)
```
{
  (i=1 ; i≤N ; i++)
  {
     (j=1 ; j≤N ; j++)
     {
        print(...)
     }
  }
```

i	j	iterations
1	[1 → N]	N
2	[1 → N]	N
⋮	⋮	+ ⋮ +
N	[1 → N]	N

Total iteration ⇒ $N * N$ ⇒ $\boxed{N^2}$ ⇒ $O(N^2)$

Q.11 void func(N)
```
{
  (i=0 ; i<N ; i++)
  {
     (j=0 ; j≤i ; j++)
     {   print(i)
     }
  }
}
```

i	j	iteration
0	[0 → 0]	1
1	[0 → 1]	2
2	[0 → 2]	3
⋮	⋮	⋮
N-1	[0 → N-1]	N

Total ⇒ $1 + 2 + 3 + \cdots + N$

$\Rightarrow \frac{N(N+1)}{2} \Rightarrow \boxed{\frac{(N^2+N)}{2}}$ $\Rightarrow \cancel{\frac{N^2}{2}} + \cancel{\frac{N}{2}} \Rightarrow O(N^2)$

Break : 10:25 pm

Doubts

i=0	j=0	(0,0)	1 +
i=1	j=0, j=1	(1,0), (1,1)	2 +
i=2	j=0, j=1	(2,0), (2,1)	3

$j=2$ $\{2,2\}$

$\vdots$

$i = N-1$ $j = 0$, $=1$, $=2$, $\vdots$, $N-1$

$(N-1, 0)$
$(N-1, 1)$
$(N-1, 2)$
$\vdots$
$(N-1, N-1)$

$+ \vdots N$

Break over

Q.12

```
void fun(N)
{
  (i=1; i≤N; i++)
  {
    (j=1; j≤N; j=j*2)
    {
      ....
    }
  }
}
```

i	j	iterations
1	$[1 \to N]$	$\log_2 N$
2	$[1 \to N]$	$\log_2 N$
		$\vdots$
N	$[1 \to N]$	$\log_2 N$

$j \Rightarrow \underbrace{1 \to 2 \to 4 \to 8 \to 16 \to \dots N}_{\log N}$

Total iteration $\Rightarrow \boxed{N * \log_2 N} \Rightarrow O(N \log N)$

$1 \Rightarrow 1$
$2 \Rightarrow 10$
$4 \Rightarrow 100$
$8 \Rightarrow 1000$
$16 \Rightarrow 10000$
$5 \Rightarrow 101$

Q.15

```
void func(N)
{
  (i=1; i≤2^N; i++)
  {
    ....
  }
}
```

$i \to [1 \to 2^N]$

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

$\boxed{1 << N \Rightarrow 2^N}$

$\boxed{2^i \Rightarrow (1 << i)}$

$i \le (1 << N)$

$\Updownarrow$

$i \le 2^N$

$8 \Rightarrow$ Binary: 1000

$1 << 3 \Rightarrow 1000$

$32 \Rightarrow 1 << 5 \Rightarrow 100000$

Q.16 void func(N)

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

i	j	iterations
1	$[1 \rightarrow 2]$	2^1
2	$[1 \rightarrow 2^2]$	2^2
3	$[1 \rightarrow 2^3]$	2^3
⋮	⋮	⋮
N	$[1 \rightarrow 2^N]$	2^N

Total iterations $\Rightarrow (2^1 + 2^2 + 2^3 + \cdots + \cdots 2^N)$

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

G.P : $2^1, 2^2, 2^3, \ldots 2^N$

$a = 2$

$r = 2$

$N = N$

Sum of G.P. $\Rightarrow \dfrac{2 \cdot (2^N - 1)}{2-1}$

$\Rightarrow \boxed{2(2^N - 1)} \Rightarrow 2 \cdot 2^N - 2 \Rightarrow O(2^N)$

Q.17 void fun(N)

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

i	j	iterations
N	$[1 \rightarrow N]$	N
$N/2$	$[1 \rightarrow N/2]$	$N/2$
$N/4$	$[1 \rightarrow N/4]$	$N/4$
		⋮
1	$[1 \rightarrow 1]$	1

Total iterations $\Rightarrow N + \frac{N}{2} + \frac{N}{4} + \frac{N}{8} + \cdots + 1$

G.P

$$\Rightarrow N + \left[\frac{N}{2^1} + \frac{N}{2^2} + \frac{N}{2^3} + \frac{N}{2^4} + \cdots \frac{N}{2^k} \right]$$

$$\Rightarrow N + N \left(\underbrace{\frac{1}{2^1} + \frac{1}{2^2} + \frac{1}{2^3} + \cdots + \frac{1}{2^k}}_{k = \log_2 N} \right)$$

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

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

G.P

$a = \frac{1}{2}$, $r = \frac{1}{2}$, $\boxed{\text{terms} = \log_2 N}$ $\quad r != 1$

$$\text{SUM}_{GP} = \frac{a.(r^{\text{terms}} - 1)}{(r - 1)}$$

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

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

$$\text{SUM}_{GP} = \left(1 - \frac{1}{N}\right) = \left(\frac{N - 1}{N}\right)$$

$$\text{Total Sum} = N + \cancel{N}\left(\frac{N-1}{\cancel{N}}\right)$$

$$= N + N - 1$$

$$= 2N - 1 \Rightarrow O(N)$$

$$\boxed{\text{Approximate} \approx 2N \text{ iterations}}$$

For a very large value of N

$$\log_2 N < \sqrt{N} < N < N\log N < N\sqrt{N} < N^2 < 2^N < N!$$

How to write Big O

- → why?
- → what?
- → reason

saturday's session

1. Calculate iterations based on Input
2. Neglect lower order terms
3. Neglect constant co-efficient

$O(3N)$

$$f(N) = 10N^2 + 100N + 10^4 N^0 \Rightarrow O(N^2)$$

$$f(N) = 4N^2 + 3N + 10^6 \Rightarrow O(N^2)$$

$f(N) = 4N + 3N\log N + 10^6 \Rightarrow O(N\log N)$

$F(N) = 4N\log N + 3\,N\,\mathrm{sqrt}(N) + 10^6 \Rightarrow O(N\sqrt{N})$

$N\sqrt{N} > N\log N$

Doubts

AP:

$S = a,\ a+d,\ a+2d,\ a+3d, \dots\ a+(n-1)d$

$S = a+(n-1)d \quad a+(n-2)d \quad \dots \quad a+d + a$

$2S = 2a+(n-1)d \quad 2a+(n-1)d \quad \dots \quad 2a+(n-1)d$

$$\boxed{S = \frac{n}{2}\left(2a+(n-1)d\right)}$$

G.P.:

$S = a,\ \cancel{ar},\ \cancel{ar^2}, \dots\ a.\cancel{(r^{n-1})}$

$rS = \cancel{ar},\ \cancel{ar^2},\ ar^3 \dots \dots\ ar^n$

$rS - S = ar^n - a$

$$s(r-1) = a(r^n-1)$$

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

H.W

H.P. $\Rightarrow$ Harmonic