Lecture 2: Time complexity - 1.

→ Time complexity & space complexity

→ Asymptotic analysis

→ Big O

→ TLE (Time limit exceeded)

— TC-2.

Agenda:

no of itr.

Quiz1: Sum of first n natural no:- $\frac{n(n+1)}{2}$.

Quiz2: How many no are there in range [3, 10]

[→ inclusive

(→ exclusive

[3, 10] → 3, 4, 5, 6, 7, 8, 9, 10 ⇒ 8 no.

[a, b]

↓

$b-a+1$.

Ex: [3, 10]

$10-3+1=8$

[a, b)

↓

$b-a$.

Ex: [3, 10)

$10-3=7$.

(a, b)

↓

$b-a-1$.

Ex: (3, 10)

4, 5, 6, 7, 8, 9

Arithmetic progression

$\underbrace{4, 7}_{3}, \underbrace{\ , 10}_{3}, \underbrace{\ , 13}_{3}, \underbrace{\ , 16}_{3}, \underbrace{\ , 19}_{3}, \underbrace{\ , 22}_{3}$ - - - - - - - .

Generalise:

1	2.	3	4	5		n.
a	$a+d$	$a+2d$	$a+3d$	$a+4d$	- - -	$a+(n-1)d$
	$a+(2-1)d$	$a+(3-1)d$	$a+(4-1)d$			

$\underbrace{a \quad a+d}_{d} \quad \underbrace{a+d \quad a+2d}_{d}$

first term = a

common diff = d

no of terms = n.

sum of AP =

A.P :- $\boxed{a \quad a+d \quad a+2d \quad a+3d \quad \cdots\cdots \quad a+(n-1)d}$

(I) $S = a + (a+d) + \cdots\cdots\cdots + a+(n-2)d + a+(n-1)d$

(II) $S = a+(n-1)d + a+(n-2)d + \cdots\cdots + a+d + a$

$a+d+a+nd-2d$

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

$\underbrace{2a+(n-1)d.}_{n \text{ times}}$

$2S = n * 2a+(n-1)d$

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

Geometric progression

$\underbrace{2 \quad 4}_{2} \quad \underbrace{8}_{2.} \quad \underbrace{16}_{2} \quad \underbrace{32}_{2} \quad \underbrace{64}_{2.} \quad \cdots\cdots$

$\overset{1}{a} \quad \overset{2}{ar} \quad \overset{3}{ar^2} \quad \overset{4}{ar^3} \quad \cdots\cdots \quad \overset{n}{ar^{n-1}}.$

$\underbrace{a \quad ar}_{r} \quad \underbrace{\quad}_{r} \quad \underbrace{\quad}_{r}$

sum of GP :- $\dfrac{a(r^n-1)}{r-1}$ $[r>1]$.

Qu1

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

$i : [1, n]$.

$\#\, itr = n - \cancel{1} + \cancel{1}$

$\#\, itr = n.$

Qu2:

```
void fun(int n, int m) {
    for(i=1; i<=n; i++) {
        if (i%2==0) {
            print(i);
        }
    }
    for(i=1; i<=m  i++) {
        if (i%2==0) {
            print(i);
        }
    }
}
```

$\longrightarrow i : [1, n]$

$\#\, itr = n.$

$\longrightarrow i : [1, m]$

$\#\, itr = m.$

Total $itr = n + m.$

Qu3:

```
int fun(int n){
    s=0;
    for(i=0 ; i<=100; i++){
        s=s+i;
    }
    returns;
}
```

$i : [0, 100]$

$\# itr = 100 - 0 + 1$

$\# itr = 101$

Qu4:

```
void fun(int n){
    for(i=1; i*i<=n; i++){
        print("ok");
    }
}
```

$i * i <= n.$

$i^2 <= n.$

$i <= \sqrt{n}.$

$\Rightarrow \quad i : [1, \sqrt{n}]$

$\# itr = \sqrt{n} \cancel{-1} \cancel{+1}$

$\# itr = \sqrt{n}.$

Ques 6

```
[n>1]
void fun(n) {
    i = n;
    while (i > 1) {
        i = i/2;
    }
}
```

a.) $\frac{n}{2}$

b.) $\log_2 n$ ✓

c.) $2n$

d.) Infinite

e.) $(n-1)/2$

itr	i	
0	n	$\frac{n}{2^0}$
1	n/2	$\frac{n}{2^1}$
2	n/4	$n/2^2$
3	n/8	$n/2^3$
4	n/16	$n/2^4$

$i \longrightarrow n, \frac{n}{2}, \frac{n}{4}, \frac{n}{8}, \frac{n}{16}, \frac{n}{32} \text{ -------- } 1.$

Assume :- After k itr, loop breaks

Value of i at k^{th} itr $= 1 = \frac{n}{2^k}$.

$$\frac{n}{2^k} = 1$$

$$2^k = n.$$

Apply $\log_2$ both sides

$$\boxed{\log_2 2^k} = \log_2 n$$

$$k \cdot \log_2 2 = \log_2 n$$

$$k \frac{\log \cancel{2}}{\cancel{\log} 2} = \log_2 n$$

itr $= k = \log_2 n$

case 2: $n < 1$

itr $= 0$.

fun problem (Amazon)

Qu 7

```
void fun(n) {
    for(i=0 ; i<n; i=i*2) {
        print(ok);
    }
}
```

itr = infinite

$i = 0$

$i = i*2$ 0

$i = i*2$ 0

0

⋮

Qu 8

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

a) n

b) $2n$

c) $n/2$

d) 2^n

e) $\log_2 n$

Assume: After k itr, loop breaks.

when $i = n$ [loop breaks]

↳ Approximation

itr	$i = i*2$	
0	1	2^0
1	2	2^1
2	4	2^2
3	$8 = 4*2$	2^3
4	16	2^4
5	32	2^5
⋮	⋮	⋮

$i = n = 2^k$

$k = \log_2 n$

Qu.9.

```
void fun(int n){
    s=0;
    for(i=1; i<=n; i=i*4.{
      s=s+i;
    }
   returns;
}
```

itr = ?

H/w problem.

NESTED LOOPS

Q10

```
void fun(n) {
    for(i=1; i<=10; i++){
        for(j=1; j<=n; j++){
            print("Sharat");
        }
    }
}
```

Trick

i	j [1,n]	# itr
1	[1,n]	n - 1 + 1 = n
2	[1,n]	n
3	[1,n]	n
4	[1,n]	n
⋮		
10	[1,n]	n

Total itr $= \underbrace{n + n + n + n - - - - n}_{\text{10 times}}$

$= 10 * n$

Qu.11.

```
void fun(n) {
    for( i = 1; i<=n. ; i++ ) {
        for( j =1; j<=n; j++) {
            print ("Sharat");
        }
    }
}
```

Trick

i	j [1,n]	# itr
1	[1,n]	$n - 1 + 1 = n$.
2	[1,n]	n.
3	[1,n]	n.
4	[1,n]	n
⋮	⋮	⋮
n.	[1,n]	n.

$$\text{Total itr} = \underbrace{n + n + n + n \cdots n}_{n \text{ times}}$$

$$= n * n = n^2.$$

Qu12

```
void fun(n) {
    for(i=0; i<n; i++) {
        for(j=0; j<=i; j++) {
            print("Manish");
        }
    }
}
```

a) n

b) $\frac{n(n-1)}{2}$

c) $\frac{n(n+1)}{2}$

d) $\log_2 n$

e) $n/2$

i	j [0, i]	# itr
0	[0, 0]	1
1	[0, 1]	2
2	[0, 2]	3
3	[0, 3]	4
⋮	⋮	⋮
n−1	[0, n−1]	$n - \cancel{1} - 0 + \cancel{1} = n.$

Total itr $= 1 + 2 + 3 + 4 - - - - - - - - n$

$$= \frac{n \cdot (n+1)}{2}$$

Break: 8:40 AM

Qu.13

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

a). n

b) n^2

c) $\frac{n^2}{2}$

d) $n\log n$.

$j = j*2$

i	$j\ [1,n]$	# itr
1	$[1,n]$	$\log_2 n$.
2	$[1,n]$	$\log_2 n$
3	$[1,n]$	$\log_2 n$.
4	$[1,n]$	$\log_2 n$
⋮	⋮	⋮
n	$[1,n]$	$\log_2 n$

Total itr $= \underbrace{\log_2 n + \log_2 n + \log_2 n \cdots \log_2 n}_{n \text{ times}}$

$= n\log_2 n$

Qu 14

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

$\rightarrow$ it $[1, 2^n]$

$\#$ itr $= 2^n \cancel{-1} \cancel{+1}$

$= 2^n$

Qu 15

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

i	j $[1, 2^i]$	$\#$ itr
1	$[1, 2^1]$	2
2	$[1, 2^2] = [1, 4]$	4
3	$[1, 2^3] = [1, 8]$	8
4	$[1, 2^4] = [1, 16]$	16
⋮	⋮	
n	$[1, 2^n]$	2^n

Total itr $= 2 + 4 + 8 + 16 + \ldots\ldots\ldots 2^n$

(a) first term $= 2$

(r) common ratio $= 2$

terms $= n$

sum of GP $= \dfrac{a(r^n - 1)}{r - 1}$

$= \dfrac{2(2^n - 1)}{2 - 1}$

$= 2(2^n - 1)$

Qu. 16

```
void fun(n) {
  for( i=n; i>0; i=i/2) {
      for(j=1; j<=i; j++) {
          print ("Jasraj");
      }
  }
}
```

itr = 2n−1

i	j [1, i]	itr
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

Total no of itr $= n + \frac{n}{2} + \frac{n}{4} + \frac{n}{8} + \cdots\cdots 1$

$$\Rightarrow n + \left[\frac{n}{2} + \frac{n}{4} + \frac{n}{8} \cdots\cdots 1 \right]$$

$$1 = \frac{n}{n} = \frac{n}{2^{\log_2 n}}$$ hint

$$\Rightarrow n + \left[\frac{n}{2} + \frac{n}{4} + \frac{n}{8} \cdots\cdots \frac{n}{2^{\log_2 n}} \right] \longrightarrow \text{H/W}$$

Comparing terms [n is very large]

$\sqrt{n}$ $n\log n$ $\log_2 n$ n n^2 2^n $n\sqrt{n}$.

$n = 10^6$.

$\log_2 n. < \sqrt{n} < n < n\log n. < n\sqrt{n} < n^2 < 2^n$.

Q How to write Big O notation? ⟶ What, why?, Reasons — TC 2.

Step 1 calculate no of itr.

Step 2 Neglect lower order terms

Step 3 Neglect constant co-efficient.

__Examples:__

$f(n) = 10n^2 + 100n + 10^4 = O(n^2)$.

$f(n) = 4n\log n + 3n\sqrt{n} + 10^3 = O(n\sqrt{n}.)$

Thankyou ☺

Doubts

```
void fun(n) {
    for(i=1; i<=n; i++){
        for(j=1; j<=2^i, j++ j*2){
            for(k=1; k<=j; k++){
                print (ok)
            }
        }
}
```

$i=1 \longrightarrow j=1 \longrightarrow k=[1,1]$

1 itr

3 itr +

$j=2 \rightarrow k[1,2]$

2 itr

i=2

i=3

i=4

31 itr

$[1, 2^4=16]$

$j=1$	$k[1,1]$	1 itr
$j=2$	$[1,2]$	2 itr
$j=4$		4 itr
$j=8$		8 itr
$j=16$		16 itr

$n \longrightarrow \frac{n}{2^0}$

$\frac{n}{2} = \frac{n}{2^1}$ k steps

$\frac{n}{4} = \frac{n}{2^2}$

$\frac{n}{8} = \frac{n}{2^3}$

$\vdots$

1

$\frac{n}{2^k} = 1$

$\boxed{k = \log_2 n}$

$s = a + ar + ar^2 + ar^3 \ldots \ldots ar^{n-1}$ — (I)

multiply both sides by r.

$rs = ra + ar^2 + ar^3 + ar^4 \ldots ar^n$ — (II)

(II) − (I)

$rs - s = (\cancel{ar} - a) + (\cancel{ar^2} - \cancel{ar}) + (\cancel{ar^3} - \cancel{ar^2}) + (ar^4 - \cancel{ar^3})$
$\cancel{ar^{n-1}} - - a \quad (ar^n - \cancel{ar^{n-1}})$

$rs - s = ar^n - a$

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

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

for(i=0; i<n; i++){ $\qquad$ for(i=0, i<=n; i++){

$[0, n)$ $\qquad$ $[0, n]$

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

i=1, i<=n; i=i*2

n-1 | $(1, 2^{n-1}) = 2^{n-1}$

i	j (1, 2^i)	# it
1	$[1, 2^1] = [1, 2]$	2−1+1 = 2
2	$[1, 2^2] = [1, 4]$	4
3	$(1, 2^3) = (1, 8)$	8
⋮	⋮	16
		⋮
n	$[1, 2^n]$	2^n