



## Today's agenda

↳ No. of iterations

↳ Big O Notations



# AlgoPrep



## Quiz

↳ How many numbers are in range  $[3, 10]$  {corners included?}

$$\{3, 4, 5, 6, 7, 8, 9, 10\} \rightarrow 8$$

$$[a, b] = b - a + 1$$

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

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

## //log basics

$$\begin{aligned} n * 2 &= 10 \\ n + 2 &= 10 \end{aligned} \rightarrow \begin{aligned} n &= 10 / 2 \\ n &= 10 - 2 \end{aligned}$$

$$\begin{aligned} n^2 &= 27 \\ n^2 &= 16 \end{aligned} \rightarrow \begin{aligned} n &= \sqrt[3]{27} \\ n &= \sqrt{16} \end{aligned}$$

$$n^2 = 16 \rightarrow 2 = \log_n 16$$



→  $\log_{\text{base}} \text{number}$   
base of log

ex:  $2^{\text{ans}} = 10 \rightarrow \text{ans} = \log_2 10$

$\downarrow$   
 $\text{ans} = 3.33\dots$

(a & b are nos.)

\*  $\log_b a = \text{ans} \Rightarrow a = b^{\text{ans}}$   
(hit and trial for ans)

i)  $\log_2 64 = \text{ans}$

$$64 = 2^{\text{ans}}$$

$\hookrightarrow \text{ans} = 6$

ii)  $\log_3 343 = \text{ans}$

$$343 = 3^{\text{ans}}$$

$\hookrightarrow \text{ans} = 5\dots$



Properties:

$$\text{I} \log_a a^n = n$$

$$\text{II} \log_c(a*b) = \log_c a + \log_c b$$

$$\text{Ex: } \log_2 10 = \log_2(2+5) = \log_2 2 + \log_2 5$$



AlgoPrep



## Quiz

↳ No. of steps for  $N \rightarrow \frac{N}{2} \rightarrow \frac{N}{4} \rightarrow \dots \rightarrow 1$

$$\left( \left( N * \frac{1}{2} \right) * \frac{1}{2} \right) * \frac{1}{2} \dots = 1$$

$$N * \frac{1}{2^{\text{no. of steps}}} = 1$$

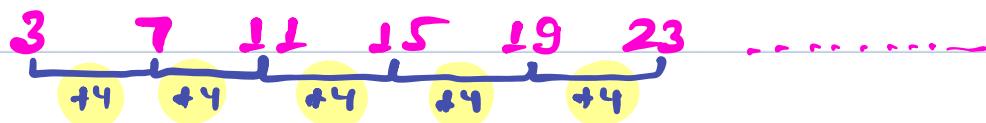


$$N = 2^{\text{no. of steps}}$$

$$\log_2 N = \text{no. of steps}$$



## A.P → Arithmetic Progression



first term =  $a$  = starting no.  $\rightarrow a = 3$

Common difference =  $d$  = diff. betw consecutive nos  $\rightarrow d = 4$

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

↳ sum of first  $n$  terms of A.P =  $\frac{n}{2} [2a + (n-1)d]$

## G.P → geometric Progression



first term =  $a$  = starting no.  $\rightarrow a = 3$

Common ratio =  $r$  = multiplier to get next no.  $\rightarrow r = 2$

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

↳ sum of first  $n$  terms of G.P =  $a \times \frac{r^n - 1}{r - 1}$



## Quiz

```
int Sum = 0;  
for (int i=1; i<=n; i++) { → [1, n] → n iterations  
    Sum = Sum + 1;  
}  
↓  
 $O(n)$ 
```

## Quiz

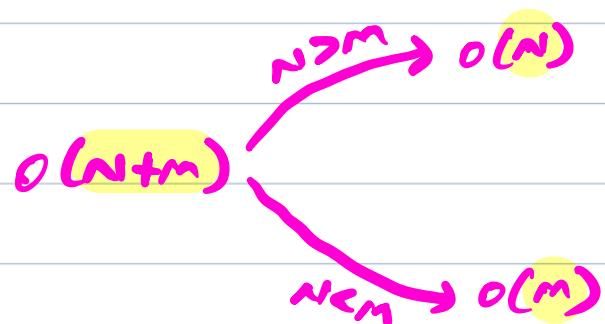
```
void func (int n, int m){
```

```
    for (int i=1; i<=n; i++) { → [1, n] = n-1+1  
        Point(i);  
    }  
    n iterations
```

```
    for (int i=1; i<=m; i++) { → [1, m] = m-1+1  
        Point(i);  
    }  
    m iterations
```

Total:  $n+m$  iterations

3





## Quiz

```
int fun (int n){  
    int s=0;  
    for (int i=0; i<=100; i++){ → [0, 100] → 100-0+1  
        s=s+i*i; → 101 iterations  
    }  
    return s; ↓ O(n)
```

3

## Quiz

# AlgoPrep

```
void fun (int n){  
    int s=0;  
    for (int i=1; i*i<=n; i++){ → [1, √n] → n-1+1  
        s=s+i*i; → √n iterations  
    }  
    return s; ↓ O(√n)
```

3



Quiz

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



→  $\log n$  iterations  
 $O(\log n)$

Quiz

# AlgoPrep

```
void fun(int n){
```

```
    int s = 0;
```

```
    for (int i = 0; i <= n; i = i + 2) { → [0, 0, 0, ...]  
        s = s + i;
```

infinite

```
    }  
}
```

3



## Quiz

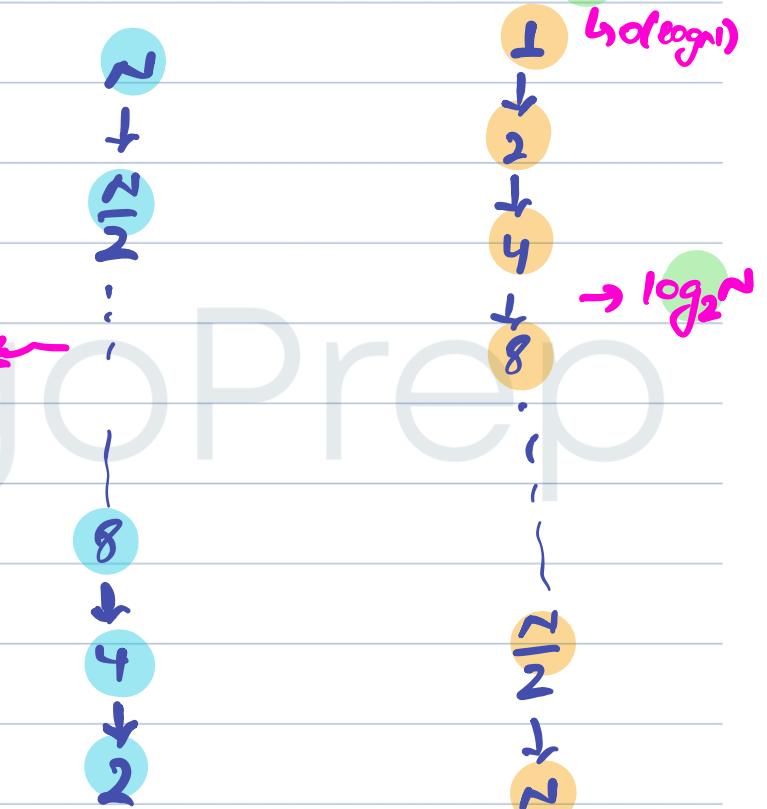
void fun(int n){

int S=0;

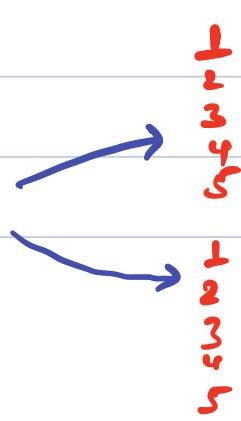
for (int i=1; i<=n; i=i\*2) {  $\downarrow \rightarrow 2 \rightarrow 4 \rightarrow 8 \dots \frac{n}{4} \rightarrow \frac{n}{2} \rightarrow n$   
 $S = S + i;$  } log n iterations

3

3



P1      P2      Break till 10:38 PM



$\rightarrow 20$



## Nested loops

### Quiz

```
void fun (int n) {
    int s = 0;
    for (int i=1; i<=10; i++) {
        for (int j=1; j<=n; j++) {
            s = s + 10;
        }
    }
}
```

i	j	Count
1	[1, n]	n
2	[1, n]	n
3	[1, n]	n
.	.	.
10	[1, n]	n
$O(n)$		$\frac{n}{10} \times n$
		Iterations

### Quiz

```
void fun (int n) {
    int s = 0;
    for (int i=1; i<=n; i++) {
        for (int j=1; j<=n; j++) {
            s = s + 10;
        }
    }
}
```

i	j	Count
1	[1, n]	n
2	[1, n]	n
.	.	.
1	1	1
n	[1, n]	n
$O(n^2)$		$n \times n$
		Iterations



## Quiz

```

void fun(int n) {
    int s=0;
    for (int i=1; i<=n; i++) {
        for (int j=1; j<=i; j++) {
            s = s + 10;
        }
    }
}

```

i	j	Count
1	[1, 1]	1
2	[1, 2]	2
3	[1, 3]	3
⋮	⋮	⋮
N	[1, N]	N

$$\frac{n^2+n}{2} = \frac{n^2}{2} + \frac{n}{2} = O(n^2) \leftarrow \frac{n*(n+1)}{2} \text{ iterations}$$

## Quiz

```

void fun(int n) {

```

```

    for (int i=1; i<=2^n; i++) {
        Point(i);
    }
}

```

3

$\rightarrow [1, 2^n] \rightarrow 2^n$  iterations  
 $O(2^n)$



## Quiz 2

```

void fun (int n) {
    int s=0;
    for (int i=1; i<=n; i++) {
        for (int j=1; j<=2^i; j++) {
            s = s + 10;
        }
    }
}

```

i	j	Count
1	[1, 2 <sup>1</sup> ]	2 <sup>1</sup>
2	[1, 2 <sup>2</sup> ]	2 <sup>2</sup>
3	[1, 2 <sup>3</sup> ]	2 <sup>3</sup>
.	.	.
N	[1, 2 <sup>n</sup> ]	2 <sup>n</sup>

$$2^1 + 2^2 + 2^3 + 2^4 + \dots - 2^n$$

$$a = 2$$

no. of terms  $\rightarrow n$

$$\sigma = 2$$

$$\text{Sum of first } n \text{ terms of G.P} = a \times \frac{\sigma^n - 1}{\sigma - 1}$$

$$= 2 \times \frac{2^n - 1}{2 - 1}$$

$$= 2 \times (2^n - 1) \text{ iterations}$$

$$2 \times 2^n - 2 = O(2^n)$$



## Comparison of iteration

$$\rightarrow n = 10^5$$

$$1 < \log n < \sqrt{n} < n < n \log n < n\sqrt{n} < n^2 < 2^n$$

Time Complexity

→ Approximate iteration Count

↳ Big O Notation

I Calculate iteration Count

→ keep highest order term

II Around + Count, neglect lower order term.

III Neglect Constants

ex: iteration count:  ~~$10n^2 + 2n + 3$~~

$$\downarrow \\ O(n^2)$$

ex:  ~~$10n^2 + 15n \log n + 2n$~~

$$\downarrow \\ O(n^2)$$



ans:  ~~$\Theta(n \log n + 3n^2 + \Delta)$~~



$O(n \log n)$

↳ The magic you are looking for is in the work you are avoiding.



AlgoPrep