Agenda

- Time complexity / Space complexity
- Asymptotic Analysis
- Big-O
- TLE

Number of iteration (why?) for different loops

1) sum of first N natural numbers → $\frac{N*(N+1)}{2}$

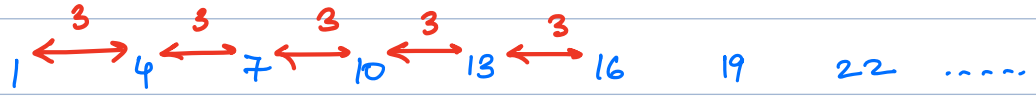
2) N → 1, no of times divide by 2 → $\log_2 N$

3) [3 10] = 8 = 10 − 3 + 1

↑ inclusive of both 3 & 10

[a b] = b − a + 1

Arithmetic progression



$d = 3$

$a \quad a+d \quad a+2d \quad a+3d \quad \ldots\ldots \quad a+(n-1)d$ ← N^{th}

$$S_n = \frac{n}{2}\left[2a + (n-1)d\right] \quad \text{or} \quad S_n = \frac{n}{2}[a + l]$$

n → no. of terms, l → last term

Geometric progression

↳ common ratio

$2 \xrightarrow{*3} 6 \xrightarrow{*3} 18 \xrightarrow{*3} 54 \xrightarrow{*3} 162 \xrightarrow{*3} \ldots\ldots$

$\frac{6}{2} = 3 \quad \frac{18}{6} = 3 \quad \frac{54}{18} = 3$

$a \quad ar \quad ar^2 \quad ar^3 \quad ar^4 \quad \ldots \quad ar^{n-1}$

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

$r < 1$

$64 \to 32 \to 16 \to 8 \to 4 \to 2 \to 1 \to 0.5 \to 0.25$

$r = 1/2$

$$S_\infty = \frac{a}{1 - r} \qquad r < 1$$

(1)

```
s = 0;
for ( int i = 0; i < N; i++)
{
    print (" Hi");
}
```

$0 \longrightarrow N-1$

$\downarrow$

N

•

```
for ( int i = 0; i < N; i++)
{
    print (" Hi");
}
```

$\} N$

$N * M$

```
for ( int i = 0; i < M; i++)
{
    print (" Hi");
}
```

$\} M$

$N + M$

•

```
for ( int i = 0; i <= 100; i++)
{
    print (" Hi");
}
```

$[0 - 100]$

$= 100 - 0 + 1$

$= 101$

-
```
s = 0;
for ( int i = 1 ; i <= 2^N ; i++)
{
    print (" Hi");
}
```

$[1 \rightarrow 2^N]$

$= 2^N - 1 + 1$

$= 2^N$

-
```
for ( int i = 1 ; i*i <= N ; i++)
{
    print (" Hi");
}
```

$i \le \sqrt{N}$

$1 \rightarrow \sqrt{N}$

$\sqrt{N}$

$i * i <= N$

$i^2 <= N$

$i <= \sqrt{N}$

-
```
for ( int i = 1 ; i <= N ; i += 2)
{ ... }
```

$1 \rightarrow N$

odd no

$N = 7$,	$1, 3, 5, 7$
$N = 9$	$1, 3, 5, 7, 9$
$N = 8$	$1, 3, 5, 7$

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

-
```
i = N;
while ( i > 1 )
{
    i = i / 2;
}
```

$i = N$

$\downarrow$

$N/2$

$\downarrow$

$N/4$

$\downarrow$

$\vdots$

$i = 1$

$\log_2 N$

$i = 0 * 2 = 0 * 2 = 0 * 2 = 0$

```
s = 0;
• for ( int i = 0; i <= N; i = i * 2 )
  {
      print ( " Hi " );
  }
```

infinite

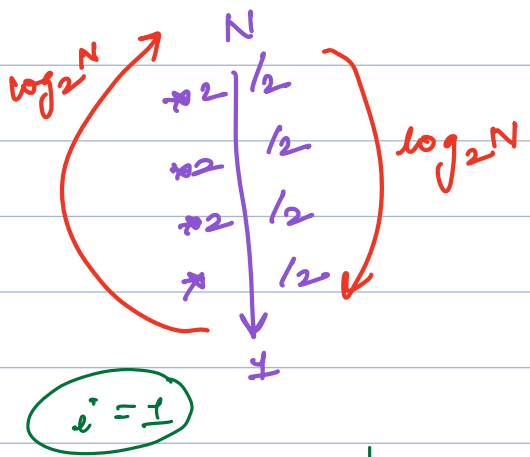
```
s = 0;
• for ( int i = 1; i <= N; i = i * 2 )
  {
      print ( " Hi " );
  }
```

~~infinite ✓ ?~~

~~$\sqrt{N}$~~

$N/2$

$\log_2 N$



$i = 1$

	+ = 2	* = 2
i	3	2
	5	4
	7	8
	9	16
	11	32
	13	64
	15	128
	$N/2$	$\log_2 N$

```
for ( int i = 1; i <= 10; i++)
{
    for ( int j = 1; j <= N; j++)
    {
        print (" Hey! ");
    }
}
```

i	j	
1	1 → N	N
2	1 → N	N
3	1 → N	N
⋮		
10	1 → N	N

$10 \times N$

```
for ( int i = 1; i <= N ; i++)
{
    for ( int j = 1; j <= N; j++)
    {
        print (" Hey! ");
    }
}
```

$N \times N$

```
for( int i=0; i<N; i++)
{
    for( j=0; j<=i ; j++)
    {
        print("Hi");
    }
}
```

i	j	
0	[0-0]	1
1	[0-1]	2
2	[0-2]	3
3	[0-3]	4
		⋮
N-1	[0-N-1]	N

$N \times (N+1)/2$

$(N^2+N)/2$

```
for ( int i = 1; i <=N ; i++)
{
    for ( int j = 1; j<=N; j*=2)
    {
        print ("Hey!");
    }
}
```

$1 \to N$

$\log_2 N$

$N * \log_2 N$

i	j	
1	$1 \to N$	$\log_2 N$
2	$1 \to N$	$\log_2 N$
3	$1 \to N$	$\log_2 N$
⋮		
N	$1 \to N$	$\log_2 N$

$N \log_2 N$

```
for ( int i = 1; i <= N ; i++)
{
    for ( int j = 1; j <= 2^i ; j++)
    {
        print (" Hey! ");
    }
}
```

i	j	
1	$[1-2]$	2
2	$[1-4]$	4
3	$[1-2^3]$	2^3
4	$[1-2^4]$	2^4
⋮		
N	$[1-2^N]$	2^N

$2+2^2+2^3+2^4 \cdots 2^N$ → GP

$a=2, r=2$

$$\frac{a(r^n-1)}{r-1} = \frac{2*(2^n-1)}{2-1} = 2(2^n-1)$$

```
for ( int i = 1; i <= N ; i++)
{
    for ( int j = 1; j <= N ; j = j + i)
    {
        print (" Hey! ");
    }
}
```

i	j	
1	$1 \to N$	N
2	$1 \to N$	N/2
3	$1 \to N$	N/3
4	$1 \to N$	N/4
⋮		
N	$1 \to N$	N/N = 1

$$= N + N/2 + N/3 + N/4 + N/5 + 1$$

$$= N\left[1 + 1/2 + 1/3 + 1/4 \cdots\right]$$

$$\approx N \log N$$

$$\int_{x=1}^{N} \frac{1}{x} = \ln x \qquad \sum 1/x = \log x$$

```
int i = 1;

while( i < N)          // log2 N
{
    int j = N;
    while ( j > 0)
    {
        j = j/2;
    }
    i = i * 2;
}
```

$\log_2 N$

$\log_2 N * \log_2 N$