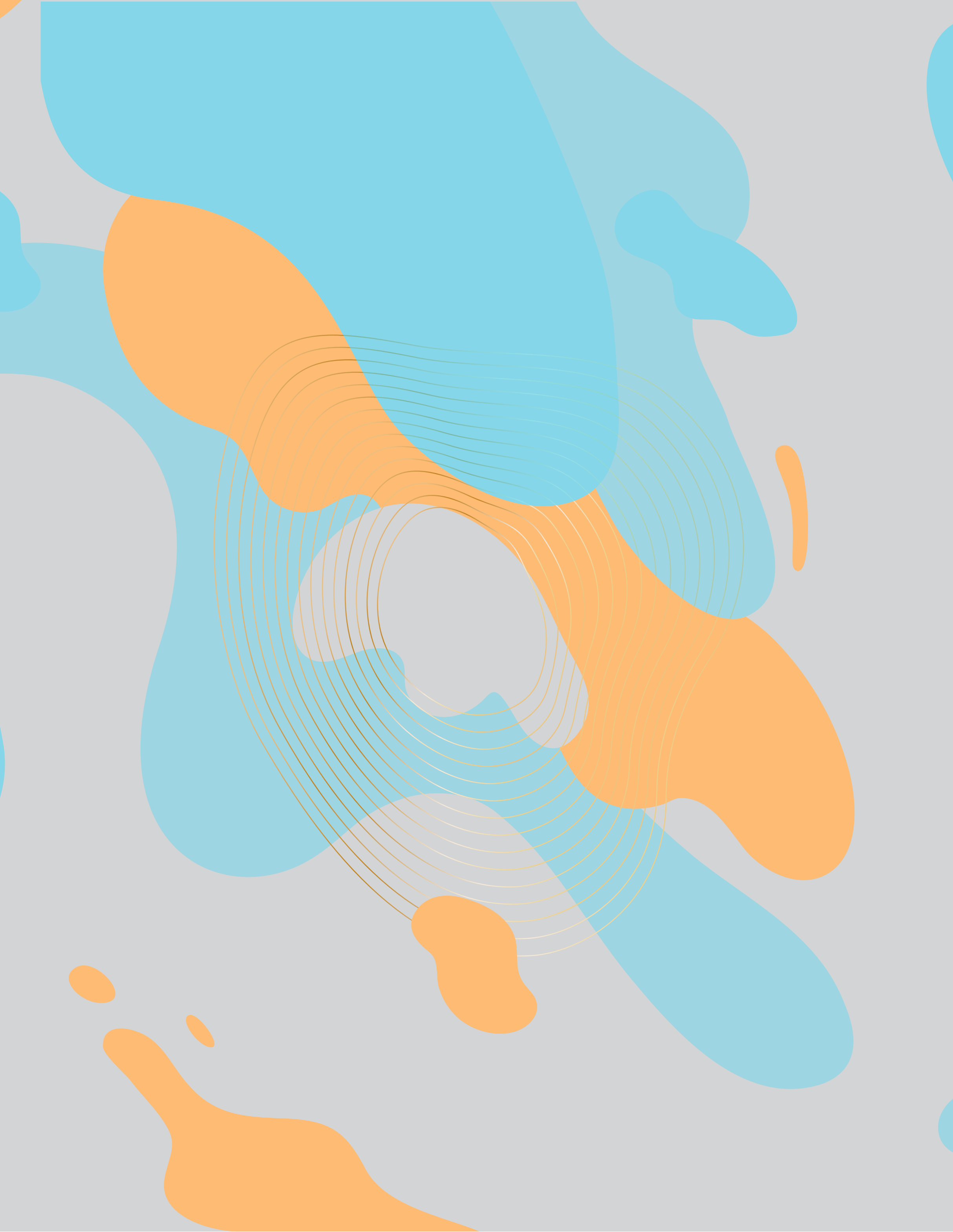

Time Complexity

In class

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

< Question > : Given a positive integer N. How many times do we need to divide it by 2 until it reaches 1?

$\Rightarrow \quad \frac{N}{2} \times \frac{1}{2} \times \frac{1}{2} \times \dots = 1$

$\Rightarrow \quad \frac{N}{2^x} = 1 \quad \Rightarrow \quad 2^x = N$

Taking log on both sides

$\log_2 2^x = \log_2 N \quad \Rightarrow \quad \boxed{x = \log_2 N}$

Quiz- 1

$\Rightarrow \quad \frac{N}{2^K} = 1 \quad \Rightarrow \quad K = \log_2 N$

N>0

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

Instructor

$N \rightarrow \frac{N}{2^1} \rightarrow \frac{N}{2^2} \rightarrow \frac{N}{2^3} \rightarrow \frac{N}{2^4} \rightarrow \text{----} \; 1$

K steps.

after K steps, loop will stop.

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

$[\log_2 N = K]$

$\therefore$ iterations $= \log_2 N$

Quiz- 2

```
for(i=1; i<N; i=i*2){
        -------------
}
```

$\Rightarrow$ $\underline{i = 1}$

$1, \quad 2, \quad 2^2, \quad \ldots \quad n \text{ times} = N$

$(2^0) \quad (2^1) \quad (2^2)$

$2^n = N$

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

Instructor

$i=1 \xrightarrow{\times 2} 2 \xrightarrow{\times 2} 4 \xrightarrow{\times 2} 8 \xrightarrow{\times 2} 16 \xrightarrow{\times 2} 32$

$\overleftarrow{/2} \quad \overleftarrow{/2} \quad \overleftarrow{/2} \quad \overleftarrow{/2} \quad \overleftarrow{/2}$

$i=1 \rightarrow 2^1 \rightarrow 2^2 \rightarrow 2^3 \rightarrow 2^4 \rightarrow 2^5 \; --- \; \underline{2^k}$

$2^k = N \Rightarrow [K = \log_2 N]$

Quiz- 3

$\underline{N \geq 0}$

```
for(i=0; i≤N; i=i*2){
        ------------
}
```

$\Rightarrow$ Since i starts from '0', it will remain 0 whatsoever be multiplied any number of times

$\Rightarrow$ Infinite iterations

$0, \quad 0 \times 2 = 0, \quad 0 \times 2 = 0, \quad \ldots \quad \infty$

Quiz- 4

```
for(i=1; i≤10; i++){
    for(j=1; j≤N; j++){
        ----------------
    }
}
```

$\Rightarrow$

i	j	iterations
1	[1, N]	N
2	[1, N]	N
3	[1, N]	N
⋮		⋮
10	[1, N]	N

$\Sigma = N + N + \ldots$ 10 times

$= 10N$

With slight variation ...

```
for( i=1; i≤10; i++){
    for( j=i; j≤N; j++){
        ________
    }
}
```

$\Rightarrow$

i	j	iterations
1	[1, N]	N
2	[2, N]	N − 1
3	[3, N]	N − 2
⋮	⋮	⋮
10	[10, N]	N − 9

$$\Sigma \text{ iterations} = N + (N-1) + (N-2) + \ldots + (N-9)$$

$$= 10N - (1 + 2 + \ldots + 9)$$

$$= 10N - \frac{9 \times \cancel{10}^{5}}{\cancel{2}} \Rightarrow \boxed{10N - 45}$$

Quiz- 5

```
for(i=1; i≤N; i++){
    for(j=1; j≤N; j++){
        ----------------
    }
}
```

⟹

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

$\sum \text{iterations} = N + N + \ldots \text{ N times} \Rightarrow \boxed{N^2}$

Quiz- 6

```
for(i=1; i≤N; i++){
    for(j=1; j≤N; j=j*2){
        ----------------
    }
}
```

⟹

i	j	iterations
1	[1, N]	$\log_2 N$
2	[1, N]	$\log_2 N$
3	[1, N]	$\log_2 N$
⋮		
N	[1, N]	$\log_2 N$

$1, 2, 2^2 \ldots k \text{ times}$

$\Rightarrow 2^k = N$

$k = \log_2 N$

$\sum \text{iterations} = \log_2 N + \log_2 N \ldots (\text{N times}) = \boxed{N \log_2 N}$

Quiz- 7

```
for(i=1; i≤4; i++){
    for(j=1; j≤i; j++){
        //print(i+j)
    }
}
```

$\Rightarrow$

i	j	iter.
1	[1, 1]	1
2	[1, 2]	2
3	[1, 3]	3
4	[1, 4]	4

$$\Rightarrow \sum_{i=1}^{4} = \frac{\cancel{4}^{2} \times 5}{\cancel{2}} = \underline{\underline{10}}$$

Quiz- 8

```
for(i=1; i≤N; i++){
    for(j=1; j≤i; j++){
        //print(i+j)
    }
}
```

$\Rightarrow$

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

$$\sum_{i=1}^{N} = \boxed{\frac{N(N+1)}{2}}$$

Quiz- 9

```
for(i=1; i≤N; i++){
    for(j=1; j≤2^i; j++){
        ------------------
    }
}
```

$\Rightarrow$

i	j	iterations	
1	$[1, 2^1]$	2	(2^1)
2	$[1, 2^2]$	4	(2^2)
3	$[1, 2^3]$	8	(2^3)
⋮	⋮	⋮	
N	$[1, 2^N]$	2^N	

$\sum$ iterations $= 2^1 + 2^2 + 2^3 + \dots 2^N$ (G.P with (r = 2))

$$\Rightarrow \frac{a(1-r^N)}{1-r} = 2\frac{(1-2^N)}{1-2} \Rightarrow \boxed{2(2^N - 1)}$$

Big O Notation

```
for(int i=1; i≤N; i++){
    if(i%2!=0){
        c=c+1;
    }
}
```

$\Rightarrow$ $\sum$ iterations $= N$ $\Rightarrow$ $O(N)$

```
for(int i=1; i≤N; i=i+2){
    c=c+1;
}
```

$\Rightarrow$ 1, 3, 5, ... untill N $\Rightarrow$ Arithmetic Progressⁿ

$N =$ last Term of AP $\Rightarrow$ $a + (n-1)d$

no. of terms

$N = 1 + (n-1)2$

$N = 2n - 2 + 1 = 2n - 1$

$$\Rightarrow n = \frac{N+1}{2} \Rightarrow O(N)$$