

20.24. LOG N COMPLEXITY

EXAMPLE 1

```
for( $i = 1; i \leq n; i = i * 1$ ){  
     $c = c + 1$ ;  
}
```

ANSWER

This is a perception building as we know the increment will go from 1 to n as $1^n = 1$.

$$1^n \Rightarrow \log_1(1) = n = 1 \text{ i.e } \log_a a = 1$$

Similarly,

```
for( $i = 1; i \leq n; i = i * 2$ ){  
     $c = c + 1$ ;  
}
```

ANSWER

Here $c = c + 1$ prints 1 time , then i will be $1 \times 2 = 2$,
then $2 \times 2 = 4$, $4 \times 2 = 8 \dots$ till $2^k \geq n$
i. e. number of iteration will be 2^k but the iteration
will not be greater than that of n .

i	$i \times 2$	n
1	$1 \times 2 = 2$	n
2	$2 \times 2 = 4$	n
...
2^{k-1}	$2^{k-1} \times 2 = 2^k \geq$	n

THEREFORE,

$$2^k \geq n$$

$$k = \log_2 n \text{ or } \log n$$

Where k is number of iterations that will run
upto n .

Now say $n = 5$:

$$k = \log_2 5 = 2.3$$

BUT THE ACTUAL ITERATIONS THAT WE HAVE IS:

1, 2, 4 i.e. 3 times.

HENCE EITHER WE DO: $\lfloor \log n \rfloor + 1$ or $\lceil \log n \rceil$

floor(logn) + 1 or ceil(logn)

HENCE NUMBER OF ITERATIONS:

$\lfloor \log n \rfloor + 1$ or $\lceil \log n \rceil$

AND COMPLEXITY STANDS AS:

$O(\lfloor \log n \rfloor + 1)$ or $O(\lceil \log n \rceil) = O(\log n)$

EXAMPLE 2

```
for( $i = 1; i \leq n; i = i * 3$ ){  
     $c = c + 1$ ;  
}
```

Number of iterations: $\lfloor \log_3 n \rfloor + 1$ or $\lceil \log_3 n \rceil$

COMPLEXITY:

$O(\lfloor \log_3 n \rfloor + 1)$ or $O(\lceil \log_3 n \rceil) = O(\log n)$