Time & Space Complexity

What is time Complexity ?

Amount of time taken by an algorithm to run as a function of length of input.

→ Time taken by CPU to process the desired input.

```
for (int i=0; i<N; i++)
{
    // operation
    Cout << "Hello";
}
```

No of operation

3

$f(n) \propto n$

Input

Note → How much time will CPU will take to perform operation on the function as per the input given

→ CPU operation

$$T.C \rightarrow O(n)$$

Why to study T & S Complexity?

1. Good computer engineer always thinks about the Complexity of the code written by him.
2. Resources are limited.
3. Measure algorithm to make efficient programs.
4. Asked by interviewer after every solution you give.

ample

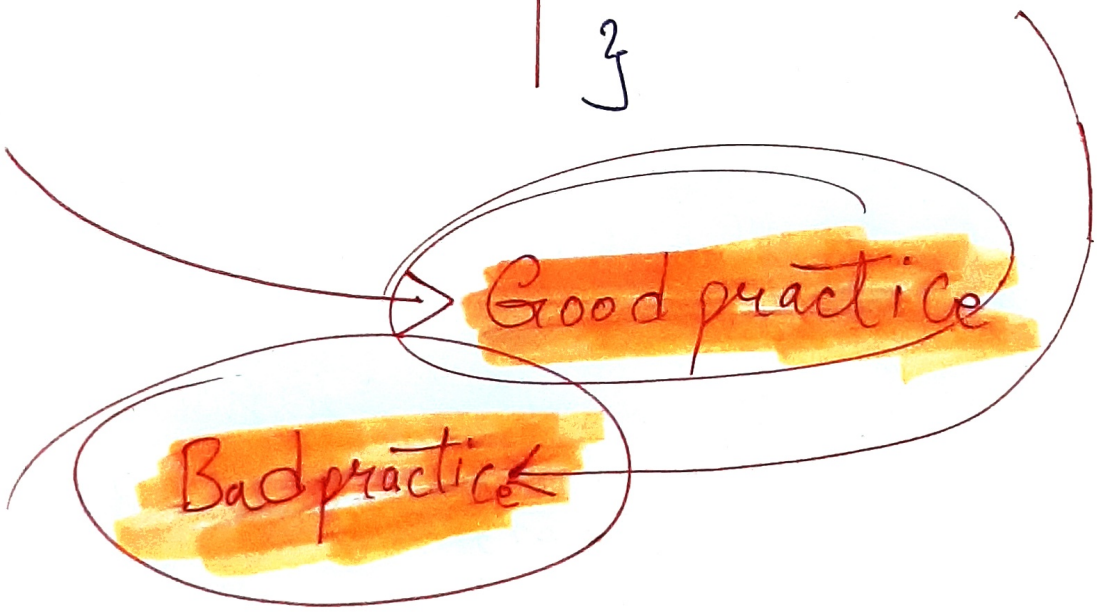
print counting

example 1

```
for(int i=0; i<N; i++)
{
    cout << i+1;
}
```

example 2

```
for(int i=0; i<N; i++)
{
    int K = N;
    While(K--);
    Cout << i+1;
}
```



Note → The purpose of time complexity is to measure which Algorithm is best and optimized.

What is Space Complexity ?

1. Amount of space taken by an Algorithm to run as a function of length of input.

Unit to represent Complexity

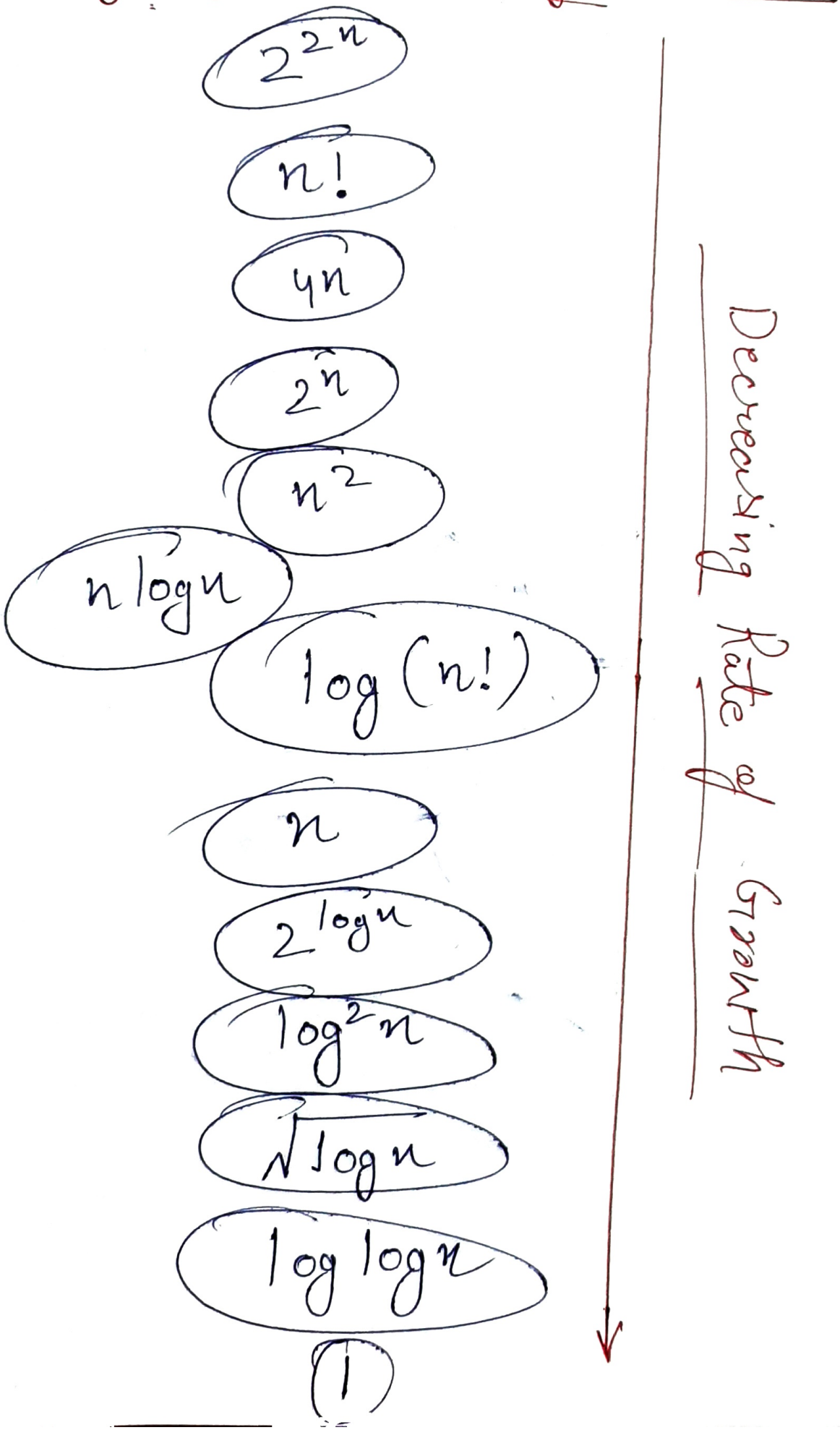
1. **Big O:** Upper bound → Worst
   → Average
2. **Theta Θ:** Average Case
3. **Omega Ω:** Lower bound → Best

Big O: Complexities

1. Constant time : $O(1)$
2. Linear time : $O(n)$
3. Logarithmic time : $O(\log N)$
4. Quadratic time : $O(N^2)$

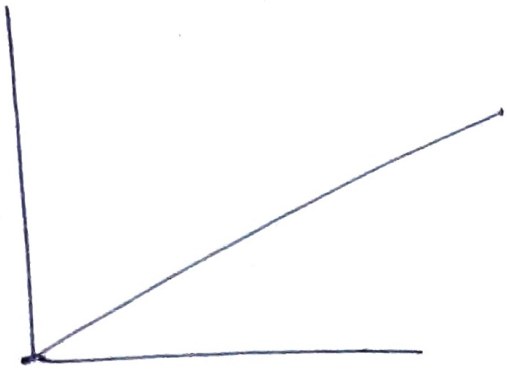
Cubic time: $O(N^3)$

Commonly used Rate of Growth

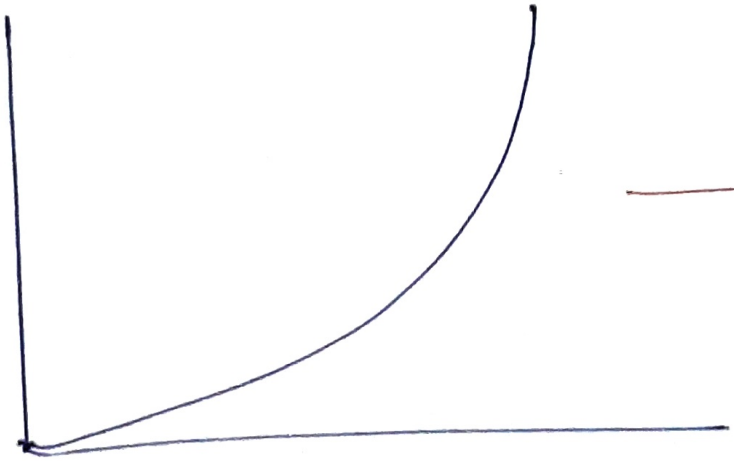


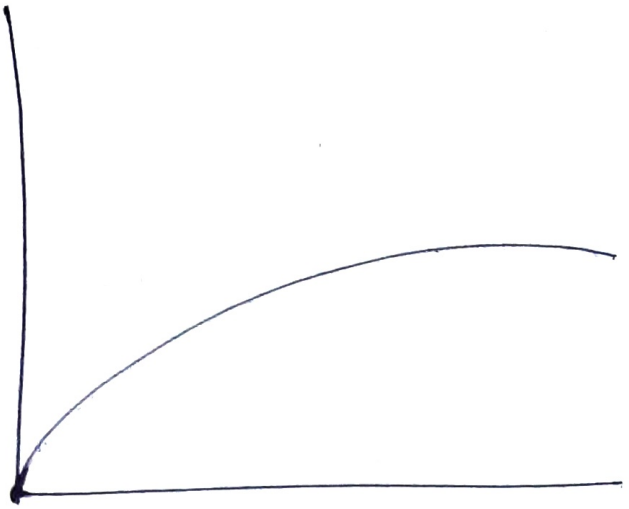
ample

1. n-axis → Input
2. y-axis → No of cpu operations

1.  → O(n)

2.  → O(1)

3  → O(n²)



$\rightarrow O(\log n)$

Questions

1) $f(n) = 2n^2 + 3n \Rightarrow 2n^2$

$O(n^2)$

2) $f(n) = 4n^4 + 3n^3 \Rightarrow 4n^4$

$O(n^4)$

3) $f(n) = N^2 + \log n \Rightarrow N^2$

$O(n^2)$

4) $f(n) = O(200) \Rightarrow O(200)$

$= O(1)$

Ex.

$> f(n) = N/4 = cN/4$

$O(N)$