Agenda

1. Log Basics + Iteration Problems
2. Comparing Iterations using Graph
3. Time Complexity - Definition and Notations (Asymptotic Analysis - Big O)
4. Why do we get TLE ?

Class starts at 7:05 IST

CREATING A ROUTINE

Routines can improve overall health, well-being, and productivity.

There are many starting points to get into a routine, but the number one rule is to make it work for you. A routine is unique to everyone, just because a routine works for your friend doesn't mean it's the best for you to take.

SCALER

Log Basics

Q. What is the meaning of LOG ?

Inverse of exponential function.

Q. How to read the statement "$\log_b(a)$"?

Base

To what value should we raise "b" such that we get "a".

If $\log_b a = c$ then $b^c = a$

1. $\log_2 64 =$

$2^x = 64$

$2^x = 2^6 \Rightarrow x = 6$

2. $\log_3 27 =$

$3^x = 27$

$3^x = 3^3 \Rightarrow x = 3$

3. $\log_2 32 =$

$2^x = 32$

$2^x = 2^5 \Rightarrow x = 5$

4. $\log_2 10 =$

$2^x = 10 \Rightarrow x = 3$

$2^3 < 10 < 2^4$

8 16

5. $\log_2 40 =$

$2^x = 40 \Rightarrow x = 5, ---$

$2^5 < 40 < 2^6$

32 64 $\Rightarrow x = 5$

Generalising :

$\log_2 N = k$

$N = 2^k$

6. $\log_2 2^6 =$ $2^x = 2^6$

$x = 6$

7. $\log_3 3^5 =$ $3^x = 3^5$

$x = 5$

Generalising :

$$\log_a a^N = N$$

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

N = 100

$100 \longrightarrow 50 \longrightarrow 25 \longrightarrow 12 \longrightarrow 6 \longrightarrow 3 \longrightarrow 1$

ans : 6

N = 324

$324 \longrightarrow 162 \longrightarrow 81 \longrightarrow 40 \longrightarrow 20 \longrightarrow 10 \longrightarrow 5 \longrightarrow 2 \longrightarrow 1$

ans : 8

SCALER

Generalising :

$$N \rightarrow \frac{N}{2} \rightarrow \frac{N}{4} \rightarrow \frac{N}{8} \; - - - - \; - - - \; 1$$

$$\frac{N}{2^0} \rightarrow \frac{N}{2^1} \rightarrow \frac{N}{2^2} \rightarrow \frac{N}{2^3} \; - - - - - - \; \frac{N}{2^k}$$

k steps to reach 1 from N

$$N/2^k = 1 \Rightarrow N = 2^k \Rightarrow \log_2 N = k$$

QUIZ

N = 9

$$9 \rightarrow 4 \rightarrow 2 \rightarrow 1$$

$$\log_2 9 \Rightarrow 2^3 < 9 < 2^4$$
$$\Rightarrow 8 < 9 < 16$$
$$\Rightarrow x = 3.\ldots$$
$$\Rightarrow x = 3$$

ans : 3

QUIZ

N = 27

$$27 \rightarrow 13 \rightarrow 6 \rightarrow 3 \rightarrow 1$$

$$\log_2 27$$
$$2^4 < 27 < 2^5$$
$$2^4 < 2^x < 2^5$$

$$x = 4$$

ans: 4

SCALER

Quiz- 3

How many iterations will be there in this loop ?

```
N > 0
i = N;
while(i > 1)
{
    i=i/2;
}
```

$N \rightarrow \frac{N}{2} \rightarrow \frac{N}{4} \rightarrow \frac{N}{8} \; - - - - - \; 1$

ans : $\log_2 N$

$O(\log_2 N)$

Quiz- 4

How many iterations will be there in this loop

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

$N = 32$

$32 \underset{\times 2}{\overset{\div 2}{\rightleftarrows}} 16 \underset{\times 2}{\overset{\div 2}{\rightleftarrows}} 8 \underset{\times 2}{\overset{\div 2}{\rightleftarrows}} 4 \underset{\times 2}{\overset{\div 2}{\rightleftarrows}} 2 \underset{\times 2}{\overset{\div 2}{\rightleftarrows}} 1$

ans : $\log_2 N$

$O(\log_2 N)$

SCALER

Amazon

Quiz- 5

How many iterations will be there in this loop ?

```
N>=0
for(i=0; i<=N; i = i*2)
{
    ...
}
```

i_{before}	iteration	i_{after}
0	1	$0*2=0$
0	2	$0*2=0$
0	3	0
0	4	0
	⋮	
0	∞	0

ans : ∞

Quiz- 6

How many iterations will be there in this loop

```
for(i=1; i<=10; i++){
    for(j=1; j<=N; j++){
    /      ......../
    }
}
```

for(i = 1 → 10)
for (j = 1 ; j ≤ N ; j++) {
| =
3 =

i	j	No of iterations
1	[1, N]	N
2	[1, N]	N
3	[1, N]	N
⋮	⋮	⋮
10	[1, N]	N

10 times

$\overbrace{N+N+N \; - - \; N}$

ans : $10*N$

$O(N)$

SCALER

Quiz- 7

How many iterations will be there in this loop

```
for(i=1; i<=N; i++){
    for(j=1; j<=N; j++){
        ...
    }
}
```

i	j	No of iterations
1	[1, N]	N
2	[1, N]	N
3	[1, N]	N
⋮	⋮	⋮
N	[1, N]	N

ans: $\underbrace{N + N + N - - - - N}_{N \text{ times}}$

$= N * N = N^2$

$O(N^2)$

Quiz- 8

How many iterations will be there in this loop

```
for(i=1; i <= N; i++){
    for(j=1; j <= N; j = j*2){
        ...
    }
}
```

i	j	No of iterations
1	[1, N] * 2	$\log_2 N$
2	[1, N] * 2	$\log_2 N$
3	⋮	⋮
N	[1, N] * 2	$\log_2 N$

ans: $\underbrace{\log_2 N + \log_2 N - - - \log_2 N}_{N \text{ times}}$

$= N * \log_2 N$

$O(N \log_2 N)$

SCALER

Quiz- 9

How many iterations will be there in this loop ?

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

i	j [1, i]	No of iterations
1	[1, 1]	1
2	[1, 2]	2
3	[1, 3]	3
4	[1, 4]	4

ans : $1 + 2 + 3 + 4 = 10$

$O(1)$

Quiz- 10

How many Iterations will be there in this loop ?

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

i	j [1, i]	No of iterations
1	[1, 1]	1
2	[1, 2]	2
3	[1, 3]	3
⋮	⋮	⋮
N	[1, N]	N

ans : $1+2+3 - - - - N$

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

$= \frac{N^2}{2} + N/2$

$O(N^2)$

SCALER

Quiz- 11

How many iterations will be there in this loop

```
for(i=1; i<=N; i++){
    for(j=1; j<=(2^i); j++)
    {
        ...
    }
}
```

i	j $[1, 2^i]$	No of iterations
1	$[1, 2^1] = [1, 2]$	2
2	$[1, 2^2] = [1, 4]$	4
3	$[1, 2^3] = [1, 8]$	8
4	$[1, 2^4] = [1, 16]$	16
⋮	⋮	⋮
N	$[1, 2^N]$	2^N

ans : $\overset{a}{2} + \overset{ar}{4} + \overset{ar^2}{8} + \overset{ar^3}{16} - - - - - \overset{ar^N}{2^N}$

$a = 2$

$r = 4/2 = 2$

Sum of N terms of GP $= \dfrac{a(r^N - 1)}{r - 1}$

$= \dfrac{2(2^N - 1)}{2 - 1}$

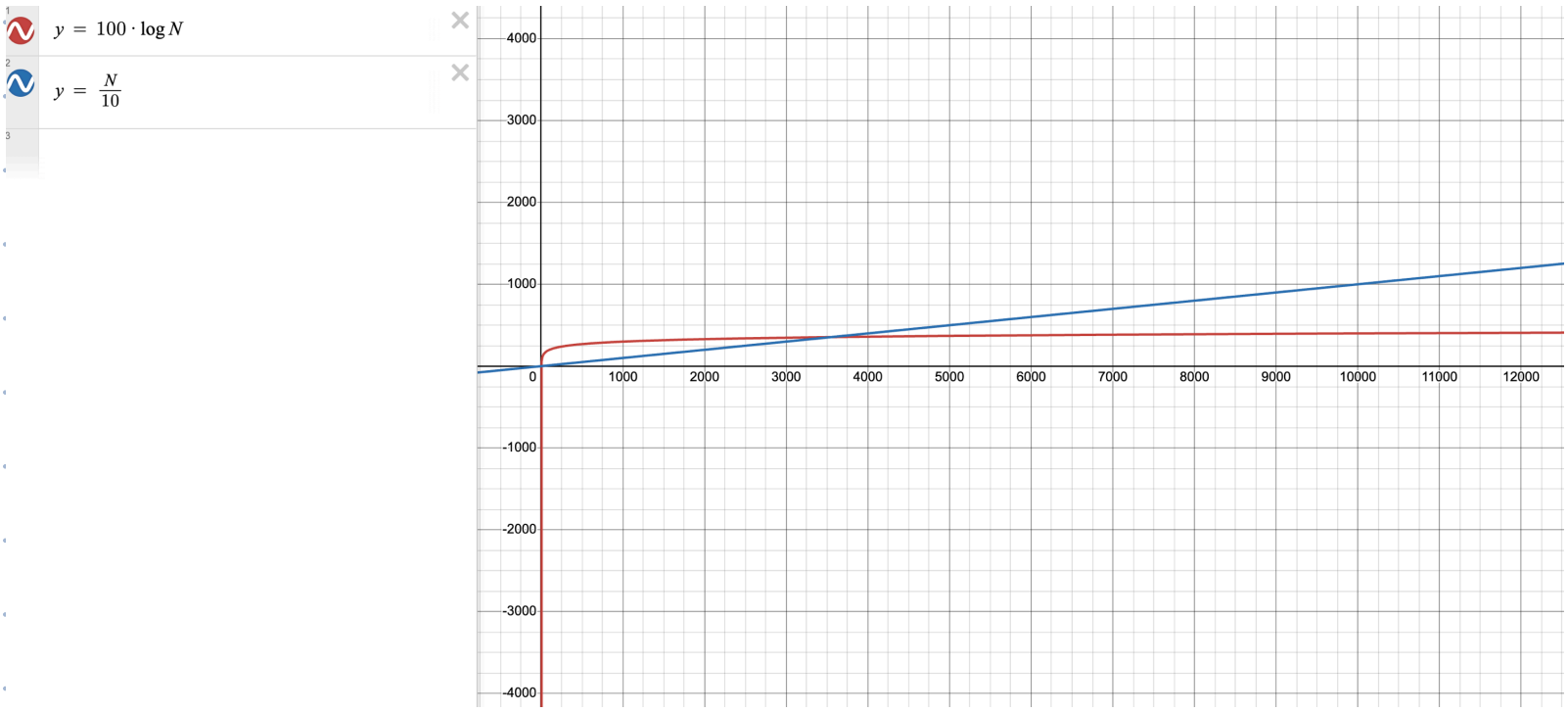
$= 2(2^N - 1)$

$2 \cdot 2^N - 2$

$O(2^N)$

Comparing Iterations of two algorithms using Graphs

	Bharadwaj's Algo (Algo1)	Daksh's Algo (Algo2)
No of iterations :	100*logN	N/10



Till 3550, algo 2 is better (BLUE one)

After 3550, algo 1 (RED) is better.

- India vs Australia (world cup final) → 6 crore
- Baby Shark YT video → 15B

SCALER

Asymptotic analysis of Algorithms

Asymptotic Analysis or Big(O) simply means analysing performance of algorithms for larger inputs.

Steps for Big O calculation are as follows:

- Calculate Iterations based on Input Size
- Ignore Lower Order Terms
- Ignore Constant Coefficients

Comparsion Order: (Valid for large value of N)

log(N) < sqrt(N) < N < N log(N) < N sqrt(N) < N^2 < N^3 < 2^(N) < N! < N^N

Using an example

N = 36

$5 < 6 < 36 < 36*5 < 36*6 < 36^2 < 36^3 < 2^{36} < 36! < 36^{36}$

What is the BigO notation time complexity of following expressions?

1) $4N^2 + 3N + 1 \rightarrow O(N^2)$

2) $4N^2 + 3N + 6 \text{ sqrt}(N) + 9 \log(N) + 10 \rightarrow O(N^2)$

3) $100 \log N \rightarrow O(\log N)$

4) $N/10 \rightarrow O(N)$

Quiz 12: $4N + 3 N \log N + 1 \rightarrow O(N \log N)$

Quiz 13: $4N \log N + 3N \text{ sqrt}(N) + 10^6 \rightarrow O(N \text{ sqrt } N)$

SCALER

Why do we ignore lower order terms?

No of Iterations → $N^2 + 10.N$

N	Total Iterations = $N^2 + 10N$	Lower Order Term $10N$	Contribution of Lower Order Term in Total Iterations = $\frac{10N}{N^2+10N} * 100$
10	$10^2 + 10 * 10 = 200$	$10 * 10 = 100$	50 %
100	$(10^2)^2 + 10.10^2 = 10^4 + 10^3$	$10.10^2 = 10^3$	~ 9%
1000	$(10^3)^2 + 10.10^3 = 10^6 + 10^4$	$10.10^3 = 10^4$	~ 1%

Conclusion: As Input size increases, contribution of lower order term decreases.

Why to neglect co-efficient / constants?

Algo 1	Algo 2	Winner
$10 \log N$	N	$10 \log N$
$100 \log N$	N	$100 \log N$
$9N$	N^2	$9N$
$10N$	$N^2/10$	$10N$

Conclusion: Growth rate of an algo remains same irrespective of the value of const.

Issues with Big(O)

Issue 1

We cannot always say that one algorithm will always be better than the other algorithm.

Algo 1 : $10^3 N \rightarrow O(N)$

Algo 2 : $N^2 \rightarrow O(N^2)$

Input Size (N)	Algo 1 ($10^3 N$)	Algo 2 (N^2)	Optimised
10	10^4	10^2	Algo 2
100	10^5	10^4	Algo 2
10^3	10^6	10^6	Both same
10^3+1	$10^3(10^3+1)$ $= 10^6 + 10^3$	$(10^3+1)^2$ $= 10^6+10^3+10^3+1$	Algo 1
10^4	10^7	10^8	Algo 1

SCALER

Issue 2

If 2 algorithms have same higher order terms, then Big O is not capable to identify algorithm with higher iterations.

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

N iterations

T.C : O(N)

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

$\frac{N}{2}$ iterations

T.C : O(N)

SCALER

Time Limit Exceeded (T.L.E)

Nishant → Amazon
(Online Assessment)

Q1 → Approach → Implement → Submit → TLE

Is it necessary to write the entire code and then test it to determine its correctness?

Can we assess the logic's viability before writing any code?

SCALER

Online Editors and Why T.L.E occurs?

- Codes are executed on online servers of various platforms such as Codechef, Codeforces, etc.

- The processing speed of their server machines is 1 GHz which means they can perform 10^9 instructions per second.

  + − * / initialisation, assignment
  < == |

- Generally, codes should be executed in 1 second.

```
int CountFactors (int N) {
    int count = 0;

    for (i = 1 ; i ≤ N ; i++) {
        if (N % i == 0) {
            count++;
        }
    }

    return count;
}
```

1 iteration → 5 instructions
N iterations → 5N "

Total instructions: 5N + 3.

Conclusion: Calculating exact no. of instructions is tedious, so we make approximation in terms of no. of iterations.

Approximation 1 : 1 iteration = 10 instructions

10^9 instruction $\longrightarrow$ 1 sec

$10^8 \times 10$ " $\longrightarrow$ 1 sec

10^8 iterations $\longrightarrow$ 1 sec

Approximation 2 : 1 iteration = 100 instructions

10^9 instructions $\longrightarrow$ 1 sec

$10^7 \times 100$ " $\longrightarrow$ 1 sec

10^7 iterations $\longrightarrow$ 1 sec

Conclusion : Our code can have $10^7 - 10^8$ iterations to be able to run in 1 sec