1) Time Complexity
2) Asymptotic analysis (Big O)
3) Big O [Formal Definition]
4) TLE (Time Limit Exceeded)

No of iterations

We start at 9:05

1) $\log_2 N$ Logarithm $7/2 = 3.5 \rightarrow 3$

$N \rightarrow N/2 \rightarrow N/4 \_ \_ \_ \_ \_ \rightarrow 1$

no of times you divided = $\log_2 N$

integer division

$7 \rightarrow 3 \rightarrow 1$ ans = 2

$9 \rightarrow 4 \rightarrow 2 \rightarrow 1$ ans = 3

$27 \rightarrow 13 \rightarrow 6 \rightarrow 3 \rightarrow 1$ ans = 4

$\log_2 2^x = x$

$2^x \rightarrow 2^{x-1} \rightarrow 2^{x-2} \rightarrow 2^{x-3} \ldots \rightarrow 2 \rightarrow 1$

2^4 $16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$ ans = 4

Q

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

$N \rightarrow N/2 \rightarrow N/4 \rightarrow N/8 \dots \rightarrow 1$

ans = $\log_2 N$

Q

```
S = 0
for (i=1 ; i ≤ N ; i=i*2) {
    S = S + i
}
```

1 2 4 8 16 32 $\log_2 N$

N

Q

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

before	after	iteration
0	0	1
0	0	2
0	0	3

Q
```
for (i=1 ; i≤10 ; i++) {
    for (j=1 ; j≤N ; j++) {
        print (i*j)
    }
}
```

i	j
1	N
2	N
3	N
⋮	⋮
10	N
	10N

iter = 10N

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

i	j
1	N
2	N
3	⋮
⋮	
N	N
	N×N = N^2

N×N = N^2

Q

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

i	j
1	log n
2	log n
⋮	⋮
N	log n

N times

N x log n

Q

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

i	j
1	1
2	2
3	3
4	4

10

Q

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

i	j
1	1
2	2
3	3
⋮	4
n	n

$1+2+3+4+\cdots+n$

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

Q

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

i	j
1	2^1
2	2^2
3	2^3
⋮	
N	2^N

$2^1+2^2+2^3+\cdots+2^N$

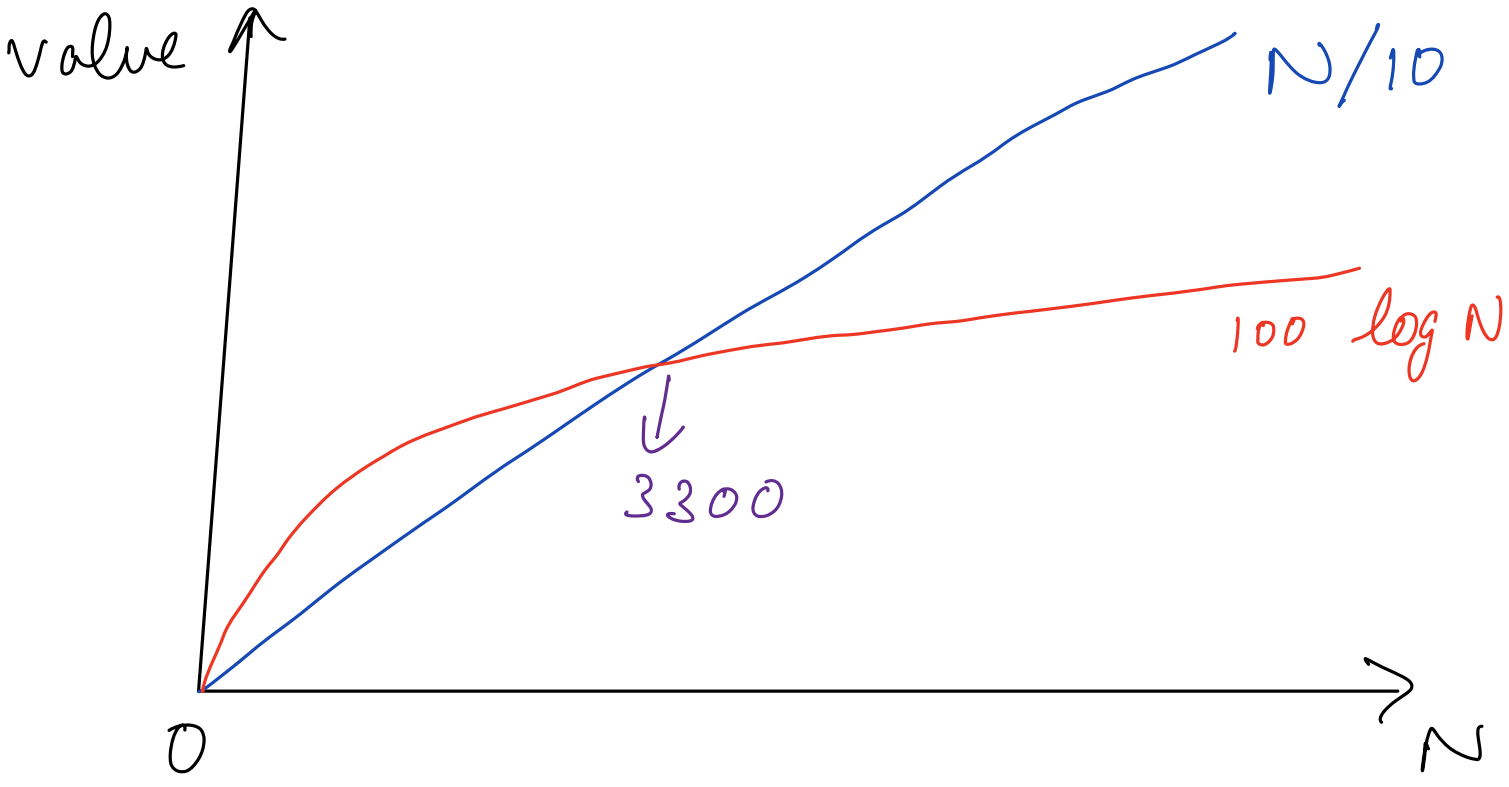
$a=2 \quad r=2$

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

Do we need to compare for small value of N or large values ?

Hotstar IND vs NZ ≈ 4 cr

Youtube Despacito ≈ 8.2 Billion



N=10 $100 \times 3 = 300$ $10^{10} = 1$

For large val $100 \log N$

Asymptotic analysis ⇒ Comparing for large inputs.

Big O

How to calculate Big O from num of iterations.

1. First find number of iterations
2. Neglect all lower order terms
3. Neglect the constant co-efficient (multiplier)

Q $4N^2 + 3N + 1$

$4N^2$

$N^2 \Rightarrow O(N^2)$

Q $4N^2 + 3N + 1331$

$O(N^2)$

Q $4N^2 + 50N$

$O(N^2)$

Q $N^2 + \log N$

$O(N^2)$

Q 10 $O(1)$

Quiz

1) $4N + 3N \log N + 1$ — $3N \log N$ — $O(N \log N)$

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

10^9 → sqrt → 30,000

10^9 → log → 30

- Why neglect lower order

	$N^2 + 10N$	
$N = 10$	$100 + 100$	50/50
$N = 10^5$	$10^{10} + 10^6$	10,000 / 1
$N = 10^8$	$10^{16} + 10^9$	1,00,00,000 / 1

Why neglect constant co-eficient

A	B	
$10 \log_2 N$	N	A
$100 \log_2 N$	$N/50$	A
$9N$	N^2	A
$10N$	N^2	A

Issues with Big O

1) Doesnt always work for smaller values

2)

A	B
$10N^2 + 2N$	$11N^2 + 5N$
$O(N^2)$	$O(N^2)$

⇒ Both algorithms are $O(N^2)$. In such cases, compare using the formula of no of iterations.

$$O(1) < O(\log N) < O(\sqrt{N}) < O(N) < O(N \log N) < O(N\sqrt{N}) < O(N^2) < O(2^N) < O(N!)$$

$N!$ → factorial of N

$1 \times 2 \times 3 \times \cdots \times N$

Time Limit Exceeded

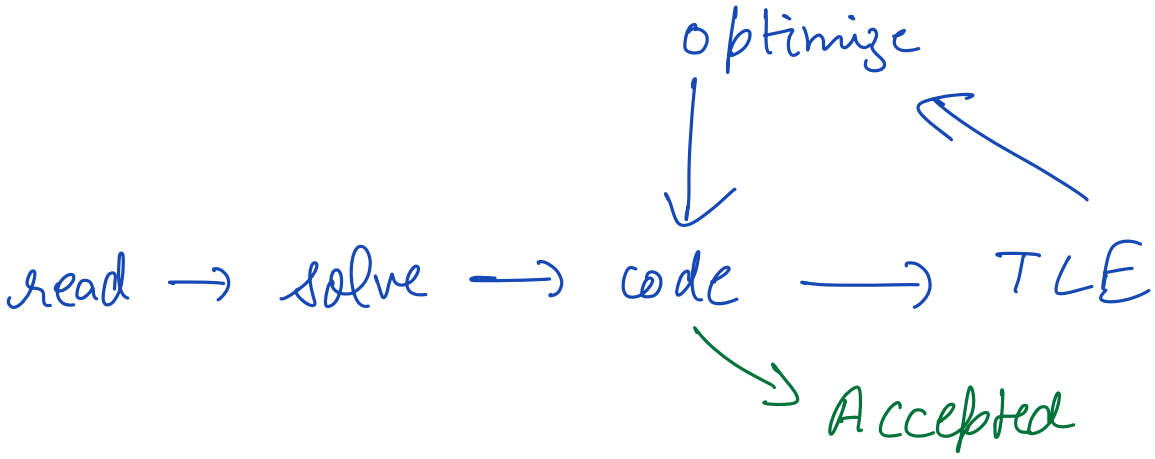
Assume Limit in Ques = 1 sec
If algo takes > 1 sec TLE

In case not given, assume limit = 1 sec
$\Rightarrow 10^8$

What to do if you get TLE
make your algo more efficient

$O(n^2) \longrightarrow O(n)$
$O(n) \longrightarrow O(\log n)$

- Online editors →
  - Can process ~ 10^8 ops /sec
  - Assume the limit = 1 sec



Why TLE occurs & constraints.

Time Limit Exceeded

Constraints $1 \le N \le 10^5$

The array size is b/w $[1, 10^5]$

Code $\Rightarrow O(N^2)$ $(10^5)^2 = 10^{10}$ iterations

10^8 iterations = 1 sec

10^{10} iterations = $10^2 \times 10^8$ = 100 sec

$N = 10^3$

$O(N)$ ✓

$O(N^2)$ $(10^3)^2 = 10^6$ ✓

$O(N^3)$ $(10^3)^3$ 10^9 ✗

$N = 100$ $O(N^3)$ ✓ 10^6

Next Class

Arrays

→ Understand
→ Observe & optimize
→ Code

i=1 i<n i+=2

i = 1 → 3 → 5 → 7 → 9 ----> <n

$n/2 \Rightarrow O(n)$

10 10 10

1) Told to TA
2) WA / Slack

$\frac{n}{2} = n \times \frac{1}{2}$

i j

1 1

1