Agenda

1. log basics + Iteration Problems
2. Comparing Iteration graphs
3. Time Complexity
4. Asymptotic Analysis (Big O)
5. TLE & Importance of constraints

Basics of Logarithm

$\log_b^a = c$

To what power should b raise such that we get a

examples

$\log_2^{64} \longrightarrow 2^? = 64 \quad [2^6 = 64]$

$\log_2^{64} = 6$

2) $\log_3^{27} = 3^? = 27 = 3^3$

3

3) $\log_5^{25} \Rightarrow 5^? = 25$

$5^2 = 2$

4) $\log_2^{32} = 2^? = 32$

5

Floor of number

$\text{floor}(\log_2^{10})$

$= 2^? = 10 = 3$

$\text{floor}(\log_2^{40})$

$= \text{floor}(5 \cdot x) = 5$

Note

$\log_a^{a^n} \longrightarrow a^? = a^n \longrightarrow n$

Question

Given a positive Integer N, how many times do we need to divide it by 2 until we get 1 (consider only Integer part)

N = 100

100 → 50 → 25 → 12 → 6 → 3 → 1

we divide 6 times to get from

100 → 1

N = 324

324 → 162 → 81 → 40 → 20 → 10 → 5 → 2 → 1

8 times division

Quiz 1

How many times we need to devide 9 by 2 till it reaches 1?

9 → 4 → 2 → 1

Explanation

Generalize

$$N \longrightarrow \frac{N}{2} \longrightarrow \frac{N}{4} \longrightarrow \frac{N}{8} \longrightarrow \cdots \longrightarrow 1$$

$$\frac{N}{2^0} \longrightarrow \frac{N}{2^1} \longrightarrow \frac{N}{2^2} \longrightarrow \frac{N}{2^3} \longrightarrow \cdots \frac{N}{2^K}$$

what is K (no. of iteration)

$$\frac{N}{2^K} = 1 \longrightarrow N = 2^K \qquad \log_2 n = K$$

How many times we need to devide 27 by 2 till it reaches 1?

$$K = \log_2 27 \qquad 2^? = 27 =$$

$$K = 4$$

Quiz 3

How many iterations will be there in this loop?

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

i before	# Iteration	i after
N	1	N/2
N/2	2	N/4
N/4	3	N/8
⋮	⋮	⋮
1	K	Stop

$$N \longrightarrow \frac{N}{2} \longrightarrow \frac{N}{4} \longrightarrow \frac{N}{8} \longrightarrow \cdots \longrightarrow 1$$

log n steps

Quiz 4

How many Iteration will be there for below loop ?

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

i before	# iteration	i after
1	1	2
2	2	4
4	3	8
⋮		
N	K	stop

example N = 32

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

$K = \log n$

Quiz 5

How many iterations will be there in below loop?

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

i before	# iteration	i after
0	1	0
0	2	0

0	3	0
:	:	:

Infinite loop

Quiz 6

How many iterations will be there in below loop ?

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

i : [1, 10]	j [1, N]	# Iteration
1	[1, N]	N +
2	[1, N]	N +
3	[1, N]	N +
:	:	:
10	[1, N]	N

Total Iteration = 10N

Quiz 7

How many iterations will be there in below loop ?

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

i [1,N]	j [1,N]	# Iteration
1	[1,N]	n times
2	[1,N]	n times
3	[1,N]	n times
⋮	⋮	⋮
N	[1,N]	n times

total Iteration = sum of all individual Iteration

$= n + n + n + \cdots + n$

$= n * n$

Quiz 8

How many iterations will be there in below loop?

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

i : [1, N]	j : [1, N] *2	# Iteration
1	[1, N]	log N
2	[1, N]	log N
3	[1, N]	log N
⋮	⋮	⋮
N	[1, N]	log N

$$\#\ \text{Iteration} = \underbrace{\log N + \log N + \cdots + \log N}_{n \text{ times}}$$

$$= n \log n$$

Quiz 9

How many iterations will be there in below loop?

```
for (int i = 1; i <= 4; i++) {
    for (int j = 1; j <= i; j ) {
        ...
    }
}
```

i : [1, 4]	j : [1, i]	# Iteration
1	[1, 1]	1
2	[1, 2]	2
3	[1, 3]	3
4	[1, 4]	4

total Iteration = 1 + 2 + 3 + 4

= 10

Quiz 10

How many iterations will be there in below loop?

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

i : [1, N]	j : [1, i]	# Iteration
1	[1, 1]	1
2	[1, 2]	2
3	[1, 3]	3
⋮	⋮	⋮
N	[1, N]	N

$$\# \text{ Iteration} = 1 + 2 + 3 + \cdots + N$$

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

Quiz 11

How many iterations will be there in below loop?

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

i [1, N]	j [1, 2^i]	# Iteration
1	[1, 2]	2 (2^1)
2	[1, 4]	4 (2^2)
3	[1, 8]	8 (2^3)
⋮	⋮	⋮ ⋮
n	[1, 2^n]	2^n (2^n)

Geometric progression

$$\# \text{ total} = 2 + 4 + 8 + \cdots + 2^n$$

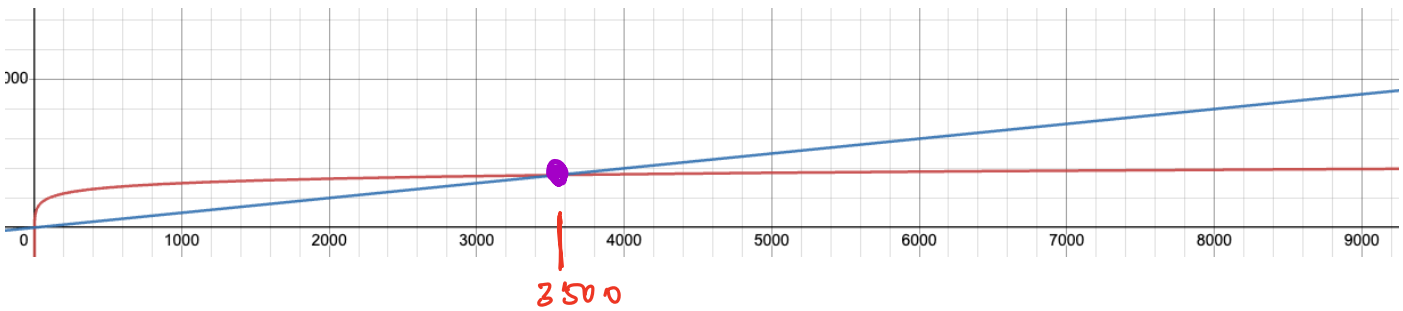
$$a = 2, \quad r = 2, \quad n = n$$

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

$$= 2(2^n - 1)$$

Comparing two algorithms using graph

	Pratham (Algo1)	Tushar (Algo2)
# Iteration	$100 \log_2 N$ ✓	$\frac{N}{10}$ ✓

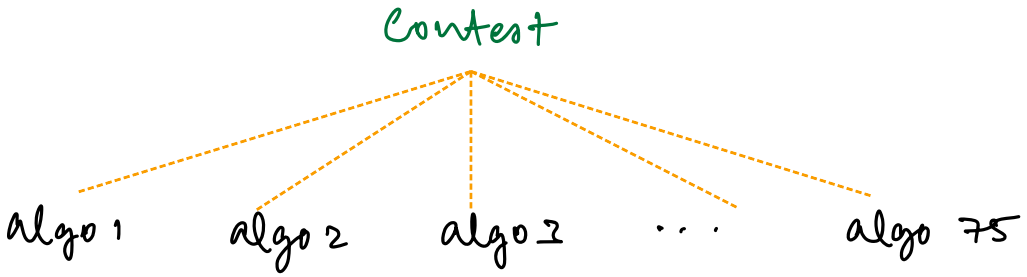


lesser iteration ⟶ lesser execution time

Observation 1	N < 3500 we prefer tushar N >= 3500 we prefer pratham
Observation 2	Pnd vs pak (5+ crores) Youtube (18 Million) Input value is usually higher in real life

Scenario

We have a contest in our batch with # of participants = 75



plotting graph is not scalable

Asymptotic analysis of algorithm

Analyse algorithm performance for larger inputs

Big O notation

Comparing multiple algorithm

a) Calculate # Iteration

b) Ignore lower order terms

c) Ignore constants

example 1 Big O: ~~100~~ $\log_2 N$

$\log_2 N$

example 2 Big O: $\frac{N}{2}$

N

---

example 3

Iteration = $4N^2 + 3N + 1$

Step 2 (Ignore lower order terms)

Step 3 (Ignore constant) $4N^2 = O(n^2)$

Comparing order

$$\log(N) < \sqrt{n} < N < N\log N < N\sqrt{n} < N^2 < N^3 < 2^n$$

$$2^n < n! < n^n$$

Comparison for N = 36

$5 \quad 6 \quad 36 \quad 36*5 \quad 36*6 \quad 36^2 \quad 3b^2 \quad 2^{36}$

Quiz 12

$F(N) = 4N + 3N\log n + 1$

Step 2 : Ignore lower order terms

$4N + 3N\log n + 1$

Step 3: Ignore coefficients

~~3~~ $N \log n = N \log n$

Quiz 13

Velmurugan

$F(n) = \cancel{4 N \log N} + 3 N\sqrt{n} + \cancel{10^6}$

Step 2: neglect lower order terms

Step 3: neglect coefficients

~~3~~ $N\sqrt{n} = O(n\sqrt{n})$

Why do we neglect lower order terms

Iteration = $N^2 + 10N$

N	N^2	$10N$	total contribution	% contribution of lower term
10	100	100	200	$\frac{100}{200} * 100 = 50\%$
100	10^4	10^3	$10^4 + 10^3$	$\simeq 9\%$
10^4	10^8	10^5	$10^8 + 10^5$	$\simeq 0.1\%$

Conclusion : as Input size increases % contribution decreases drastically

---

Why do we neglect constant coefficient ?

N	Ramesh	Vivek	Winner for large input
	$10 \log N$	N	Ramesh for larger N
	$\sqrt{n}$	$100 \log N$	Vivek is winner
	$10^6 \log n$	$\sqrt{n}$	Ramesh

while considering large Inputs constant dont make a difference

---

Issues with Big O notation

Issue 1

Claim Algo 1 < Algo 2 (Big O)

$\forall$ Inputs Algo 1 is better than Algo 2

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

Algo 2: $N^2 \longrightarrow O(n^2)$

Input	Algo 1	Algo 2
N = 10	10^4	10^2
N = 100	10^5	10^4
$N = 10^4$	10^7	10^8

Only for large inputs algo 1 does better than algo 2

Issue 2

Algo 1: $2N^2 + 4N$ Algo 2: $3N^2$

Big O : $O(N^2)$ = $O(n^2)$

Lets say $-2N^2$ in both algo

Algo 1 = $2N^2 + 4N - 2N^2 = 4N$

Algo 2 $= 3N^2 - 2N^2 = N^2$

No. of Iteration is the ultimate comparison

Time Limit Exceeded

Anuj ⟶ Tata (hiring contest)

Understand ⟶ Ideate ⟶ Code

⟵ TLE

Two question

1) Do you first have to code

HACK

Online Code executor ⟶ 1 sec

Backend servers

1 GHz

1 sec ⟶ 10^9 instructions

Smallest units in program

pseudo code

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

+1 for `i=1`, +1 for `i<=N`, +1 for `i++`, +1 for `if (N % i == 0)`, +1 for `ct = 1;`

6 instruction → 1 iteration

1 sec 10^9 → 1 sec

total iteration $\lesssim 10^8$

Approximation

1 iteration → 10 instruction

N iteration → 10^9 instruction

$10 * 10^8$ instruction → 1 sec

whenever your code

iteration $[10^7, 10^8]$

Array → Interview

1. look at constraints
2. Come up with Idea
3. Idea follows 1 sec principle
4. If not re Ideate

Importance of Constraints

Constraint $1 <= n <= 10^5$

Complexity	Iteration	> 1sec
$O(n^3)$	$(10^5)^3$	✗
$O(N^2)$	$(10^5)^2$	✗
$N \log N$	$10^5 \log 10^5$	✓

Constraints

$1 <= N <= 100$

will $O(n^3)$ work on this input

Iteration $= (100)^3 = 10^6$

totally valid

Constraints

$1 <= N <= 20$

T.C $= O(2^n)$ even this will work