Welcome !!!

## Log

$\log_a(b) = c \rightarrow a^c = b$ ✓

$\log_2(64) = 6$ $\quad 2^6 = 64$

$\log_3(81) = 4$ $\quad 3^4 = 81$

$\log_2(2^6) = 6$

$\log_3(3^4) = 4$

$\log_a(a^x) = x$ ✓

Q → Given a +ve no. N, how many times we need to divide it by 2 to reach 1.

integer division → $\frac{5}{2} = (2.5) \rightarrow 2$

N = 10 $\quad \frac{10}{2} \rightarrow \frac{5}{2} \rightarrow \frac{2}{2} \rightarrow 1$ $\quad$ Ans = 3 ✓

N = 30 $\quad \frac{30}{2} \rightarrow \frac{15}{2} \rightarrow \frac{7}{2} \rightarrow \frac{3}{2} \rightarrow 1$ $\quad$ Ans = 4

N = 9 $\quad \frac{9}{2} \rightarrow \frac{4}{2} \rightarrow \frac{2}{2} \rightarrow 1$ $\quad$ Ans = 3

N = 27 $\quad \frac{27}{2} \rightarrow \frac{13}{2} \rightarrow \frac{6}{2} \rightarrow \frac{3}{2} \rightarrow 1$ $\quad$ Ans = 4

$\frac{N}{2} \rightarrow \frac{N}{2^2} \rightarrow \frac{N}{2^3} \ldots \boxed{\frac{N}{2^K} = 1}$ (# steps)

$\Rightarrow N = 2^K$

$\Rightarrow \log_2(N) = \log_2(2^K) = K$

$\boxed{\#\text{steps} = \log_2(N)}$

floor (2.3) = 2

floor (3.6) = 3

Ans = floor ($\log_2$(N))

N = 10 floor ($\log_2$(10)) = floor (3.3..)

= 3 ✓

$2^3 = 8$

$2^4 = 16$

| int / int → int | int a = 5, b = 2<br>print (a/b) → 2 |
|---|---|

---

## Iterations

1)

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

# iterations = $\log_2$(N)  TC = O(log(N))

8 → 4 → 2 → 1

8 ← 4 ← 2 ← 1

2)

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

i = 1 → 2 → $2^2$ → $2^3$ ... $2^K = N$

$K = \log_2(N)$  TC = O(log(N))

3)

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

i = 0 * 2 = 0  No update in 'i'

⇒ # iterations = ∞

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

| i | j | # iterations |
|---|---|---|
| 1 | 1 — N | N |
| 2 | 1 — N | N |
| 3 | 1 — N | N |
| ⋮ | | |
| N | 1 — N | N |
| | | N * N |

# iterations = $N^2$ TC = $O(N^2)$

5)
```
for (i = 1 ; i <= N ; i++) {                  → N
    for (j = 1 ; j <= N ; j = j*2) {          → log2(N)
        ...
    }
}
```

# iterations = $N \log_2(N)$ TC = $O(N \log(N))$

6)
```
for (i = 1 ; i <= N ; i++) {
    for (j = 1 ; j <= 2^i ; j++) {
        ...
    }
}
```

| i | j | # iterations |
|---|---|---|
| 1 | 1 — $2^1$ | 2 |
| 2 | 1 — $2^2$ | $2^2$ |
| 3 | 1 — $2^3$ | $2^3$ |
| ⋮ | | |
| N | 1 — $2^N$ | $2^N$ |

$a = 2$
$r = 2$

$2 + 2^2 + 2^3 + \ldots + 2^N$

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

$\frac{2(2^N - 1)}{2-1} = 2(2^N - 1)$ # iterations

$TC = O(2^N)$

---

## Compare Iterations

| | Rohit | Sparsh |
|---|---|---|
| | Algo 1 | Algo 2 |
| # iterations → | $100 * \log(N)$ ✓ | $N/10$ |
| N | | |
| <= 3500 | large | small ✓ |
| > 3500 | small ✓ | large |

In Real life we have to deal with mostly large inputs.

Eg → Youtube "baby shark" → views ≈ 12 B

## Asymptotic Analysis / Big O

→ Analysing performance of algorithm over large inputs.

Steps to calculate Big O → rate of growth of function wrt input.

1) calculate iterations wrt inputs.
2) Ignore lower order terms.
3) Ignore constant coefficients.

Eg → ~~100~~ $* \log(N) = O(\log(N))$ ← log()

$N/\cancel{10} = O(N)$ ← linear

$TC = O(\#\text{iterations})$

$4N^2 + 3N - \log(N) = O(N^2)$

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

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

$\log_2(64) < \sqrt{64}$

$6 < 8$

## Why to neglect lower order terms ?

10:40 PM

| N | # iterations = $N^2$ + 10N | % contribution of 10N |
|---|---|---|
| 10 | $10^2 + 10*10 = 200$ | $\frac{10*10}{200} * 100 = 50\%$ |
| 100 | $100^2 + 10*100 = 11000$ | $\frac{10*100}{11000} * 100 = 9\%$ |
| 10000 | $10000^2 + 10*10000 = 10^8 + 10^5 \approx 10^8$ | $\frac{10^5}{10^8} * 100 = 0.1\%$ |

⇒ for large values of N, % contribution of lower order term is very less.

---

## Why ignore constant coefficient ?

Big O → rate of growth of function wrt input.

$y = x$ → linear

$y = 3x$ → linear

$y = 100x$ → linear

---

# Limitations with Big O

1)

| | Algo 1 | Algo 2 |
|---|---|---|
| # iterations → | $1000 * N^2$ | $N^2/10$ ✓ |
| Big O → | $N^2$ | $N^2$ → equally good |

2)

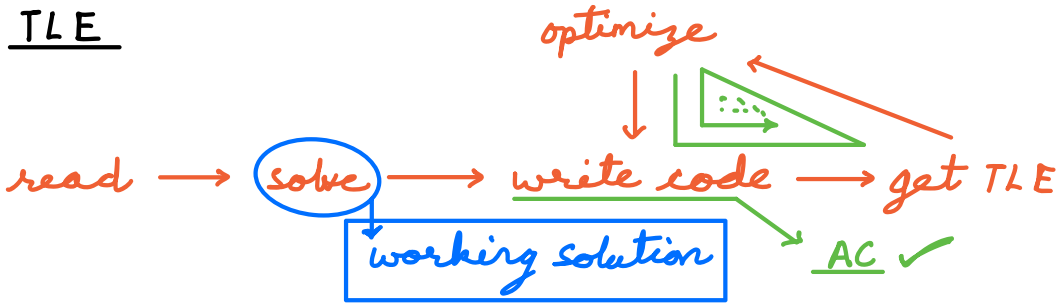
| | Algo 1 | Algo 2 |
|---|---|---|
| # iterations → | $10^6 * N\sqrt{N}$ | $N^2/10$ |
| Big O → | $O(N\sqrt{N})$ ✓ | $O(N^2)$ |

(only for very large inputs)

| N | | |
|---|---|---|
| $10^2$ | $10^6 * 10^2 * 10 = \underline{10^9}$ | $10^4/10 = \underline{10^3}$ ✓ |
| $10^4$ | $10^6 * 10^4 * 10^2 = \underline{10^{12}}$ | $10^8/10 = \underline{10^7}$ ✓ |
| $10^6$ | $10^6 * 10^6 * 10^3 = \underline{10^{15}}$ | $10^{12}/10 = \underline{10^{11}}$ ✓ |

---

## TLE




## Online Editor

processing speed ≈ 1 GHz

⇒ $10^9$ instructions per second

allowed time → 1 sec

$a = 2$

$x + y$

$(a == 1)$

1 iteration → 10 to 100 instructions (usually)

```
for ( ___ ) {
    ⋮
}
```

$10^9$ instructions → 1 sec

$10^8$ iterations → 1 sec

(1 iteration → 10 instructions)

```
for ( ___ ) {
    .
    .
    .
    .
    .
}
```

$10^9$ instructions → 1 sec

$10^7$ iterations → 1 sec

(1 iteration → 100 instructions)

→ Usually allow → $10^7$ to $10^8$ iterations per sec

Read Question → develop logic → pseudo-code (think)

TC = $O(N^2)$ — idea of TC

| Constraints | #iterations | |
|---|---|---|
| $N <= 10^3$ | $N^2 = 10^6$ | ✓ (no TLE) |
| $N <= 10^5$ | $N^2 = 10^{10}$ | X (TLE) |
| $N <= 10^4$ | $N^2 = 10^8$ | (may or may not work) |

Use constraints

| | |
|---|---|
| $1 <= N <= 10^6$ | linear solution |
| $1 <= N <= 10^3$ | $N^2$ solution |
| $1 <= N <= 10^2$ | $N^3$ solution |
| $1 <= N <= 20$ | even $2^N$ works. |