Day - 1

Saturday, 20 December 2025 12:06 PM

Rahul — SDE2 @ M2Pool ⇒ Qualcom (2022-2025) ⇒ Java

Python

RoR, Golang

- 2022 (DSA)
- Gurgaon ⇒ Guwahati
- NIT, Trichy

—SDE(I)

Q ⇒ Expectⁿ

⇒ DSA + Backend (System Design)

Q ⇒ C++ Python

↳ Java ***

↳ "Psedeo" Code

Q ⇒ Time Complexity ⇒ ⊛ Recursion

```
for (i = 1; i < n; i *= 2) {
    print(i)
}
```

Q $\boxed{\text{For / While}}$ ⇒ ✗

Q $\boxed{\text{Rec}^n}$ ⇒

DSA ⇒
- Array (1, 2D)
- Bit Manipulatⁿ
- Mathematical
- Searching (LS, BS) ⇒ Recⁿ
- Sorting (
- LL
- Stacks
- Qus

⇒ "30%"

- Heap
- Trees ⇒ 3/4
- Graph ⇒ 4/5
- DP ⇒ 4/5

GFG ⇒ Array

L.C
GFG → 4/5

LLD = { SOLID, Design Pattern, Case-studies }

HLD = { L.B, CAP, ..., Case-Studies }

Q for ($\boxed{i=0}$; $i<n$; $\boxed{i++}$) {

print(i) — O(1)

}

$\boxed{int\ i=5}$ ⇒ O(1)

$\boxed{i\ +=2}$ ⇒ O(1)

$\boxed{i=0}$ ⇒ O(1)

initialization = ①

0<n $\boxed{i<n}$ ⇒ O(1)

O(1) — i=0 $\boxed{0<n}$ ✓ — O(1)

- print(0) — O(1)
- i++ — O(1)

3·O(1) ⇒ $\underbrace{O(3\cdot 1)}_{O(1)}$ ⇒ C·O(1)

⇒ O(1)

const

⇒ O($\cancel{c}$·n)

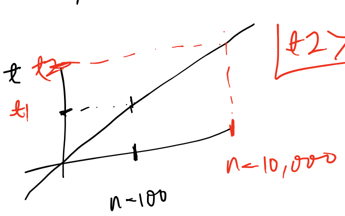
O(1) — i=1   1<n ✓   O(1)

- print(1)
- i++

O(1) — i=(n-1)   (n-1)<n   O(1)

$\underbrace{O(1)+O(1)+ \dots O(1)}_{n\ times}$ ⇒ n·O(1) ⇒ O(n)

y = x



|t2 > t1|

Q for (i=n; i>=0; i--) {

print(i)

}

$[1,2,3] \rightarrow 3$

$[0,1,2,3] \rightarrow 4$

$i \in [0, n] \rightarrow n+1$

$i = n \geq 0$ — $O(1)$

$i = (n-1) \geq 0$ — $O(1)$

⋮

$i = 3 \geq 0$ — $O(1)$

$i = 2 \geq 0$ — $O(1)$

$i = 1 \geq 0$

$i = 0 \geq 0$ — $O(1)$

$(n+1) \cdot O(1)$

$\Rightarrow O(n+1) \Rightarrow O(n)$

$[1, 3] \Rightarrow$

$[s, e] \Rightarrow [1, 3] \Rightarrow \{1, 2, 3\}$

$[2, 10] \Rightarrow \cancel{8}?\ 9$ $\qquad (R - L + 1)$

$[1, 10] \Rightarrow 10$

$(10 - 1 + 1) = 10$

$(10 - 2 + 1) = 9$ ✓

```
for (i=1; i<n; i+=2) {
    print(i)
}
```

$i = 1 \rightarrow O(1)$

$+2$

$i = 3$ — $O(1)$

$+2$

$i = 5$ — $O(1)$

⋮

$i < n$ — $O(1)$

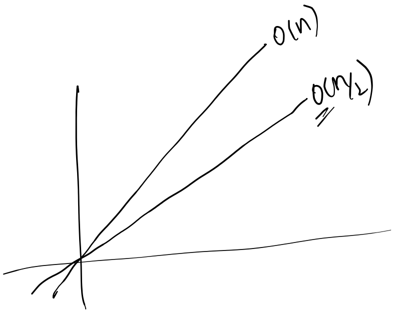
$\frac{n}{2}$ times

$= \left(\frac{n}{2}\right) * O(1)$

$\Rightarrow O\left(\frac{n}{2}\right) \Rightarrow O\left(\frac{1}{2} * n\right)$

const

$\Rightarrow O(n)$ ✓



$f(n) \leq c \cdot n$

$f(n) = O(n)$

$\boxed{f(n) \leq c \cdot n}$ ✓

Q. for (i = 0 ; i < n ; i *= 2) {
count(i)
}

⇒ ~~O($\log_2 n$)~~ ⇒ Constant.

⇒ 0

" ∞-Loop "

i = 0 < n ✓
– print (0)
– $\boxed{i = i * 2}$ ⇒ i = 0 * 2 = 0

i = 0 < n ✓
– print (0)
– i = i * 2 ⇒ i = (0) ✗

∞

Q. for (i = n ; i < 1 ; i /= 2) {
print(i)
}

C1

n >= 1 — ✗

n >= 2 ✗ | n = 1 ✗ | n = 3 ✗

C2

$\boxed{n = 0}$

i = 0   0 < 1 ✓
↳ i /= 2
⇒ i = 0

i = 0   0 < 1
↳ i /= 2
i = 0 ✗

∞

Q. for (i = 1 ; i < n ; i *= 2) {
print(i)
}

O(1)   O(1)   "k" – steps   < n

TC = O(1) · K ⇒ O(k) ⇒ O($\log_2 n$)

$i = 1 \quad 2 \quad 4 \quad 8 \; \ldots\ldots$

$2^0 \quad 2^1, \quad 2^2, \quad 2^3,$

$\boxed{2^{k-1} < n}, \quad \boxed{2^k < n}$

$O(1)$

$\boxed{2^k \geq n}$

$TC = (k) \cdot O(1) \Rightarrow O(k)$

$\Rightarrow O(\log_2 n)$

$\Rightarrow 2^k = n$

$\log_b a \Rightarrow \dfrac{\log_e a}{\log_c b}$

$\log_2 (2^k) = \log_2 n$

$\log_2 2 \Rightarrow \dfrac{\cancel{\log 2}}{\cancel{\log 2}} \Rightarrow (1)$ ✓

$k \cdot \log_2 2 = \log_2 n \quad \Rightarrow k \cdot 1 = \log_2 n$

$\boxed{k = \log_2 n}$ ✓

Q for (i=n; n>1 , i /=2) {

print(i)

}