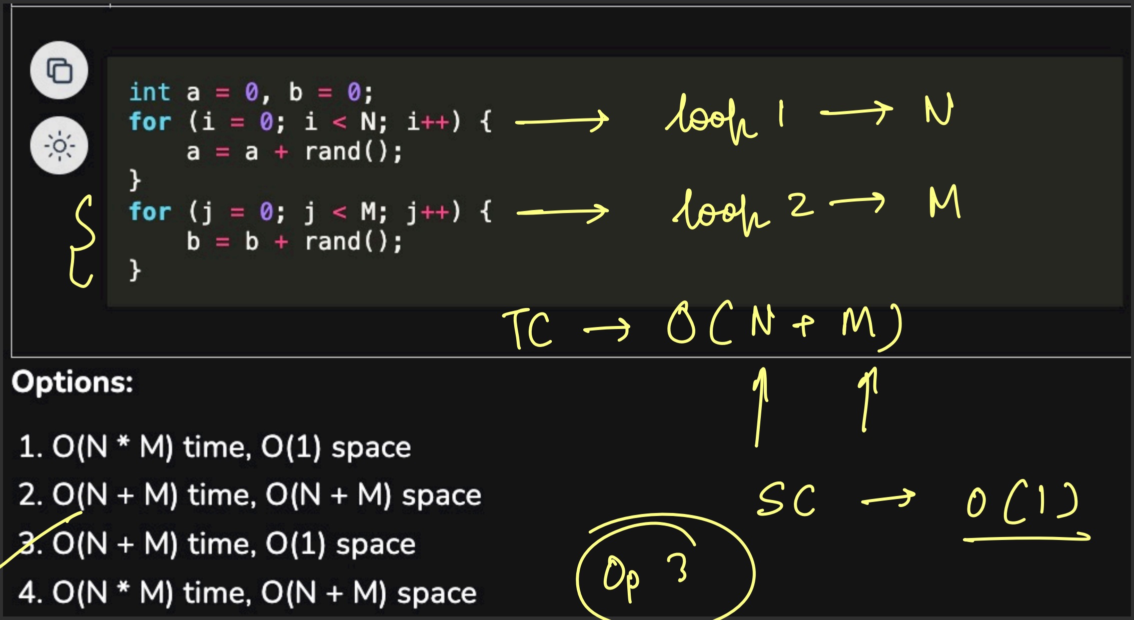
R13
Time and Space Complexity

X Data structure Operations → N



```
int a = 0, b = 0;
for (i = 0; i < N; i++) {
    a = a + rand();
}
for (j = 0; j < M; j++) {
    b = b + rand();
}
```

loop 1 → N

loop 2 → M

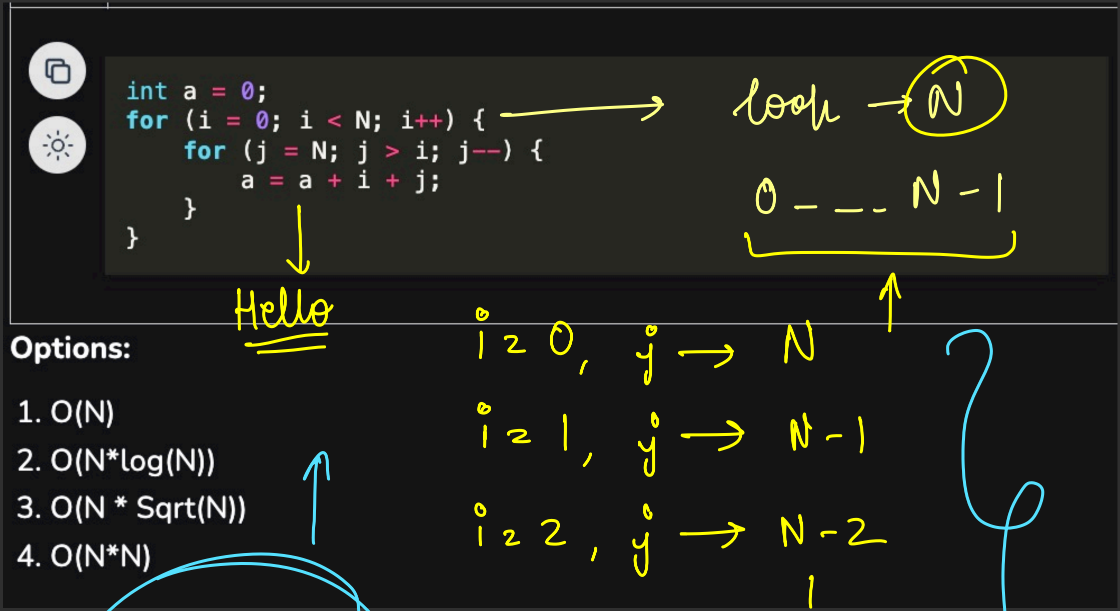
TC → O(N + M)

SC → O(1)

Options:

1. O(N * M) time, O(1) space
2. O(N + M) time, O(N + M) space
[✓ 3. O(N + M) time, O(1) space]
4. O(N * M) time, O(N + M) space

Op 3



```
int a = 0;
for (i = 0; i < N; i++) {
    for (j = N; j > i; j--) {
        a = a + i + j;
    }
}
```

Options:

1. O(N)
2. O(N*log(N))
3. O(N * Sqrt(N))
4. O(N*N)

$$\text{loop} \rightarrow N$$

$$\underbrace{0 \; --- \; N-1}$$

Hello

$$i = 0, \; j \rightarrow N$$

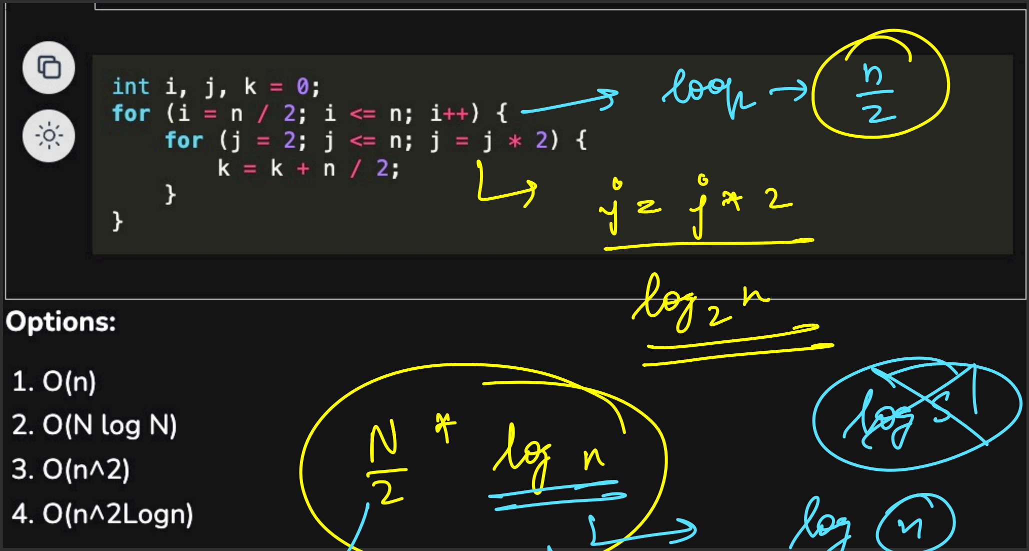
$$i = 1, \; j \rightarrow N-1$$

$$i = 2, \; j \rightarrow N-2$$

$$i = N-1 \quad j \rightarrow 1$$

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

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



```
int i, j, k = 0;
for (i = n / 2; i <= n; i++) {
    for (j = 2; j <= n; j = j * 2) {
        k = k + n / 2;
    }
}
```

Options:

1. O(n)
2. O(N log N)
3. O(n^2)
4. O(n^2Logn)

loop → $\frac{n}{2}$

$j = j * 2$

$\log_2 n$

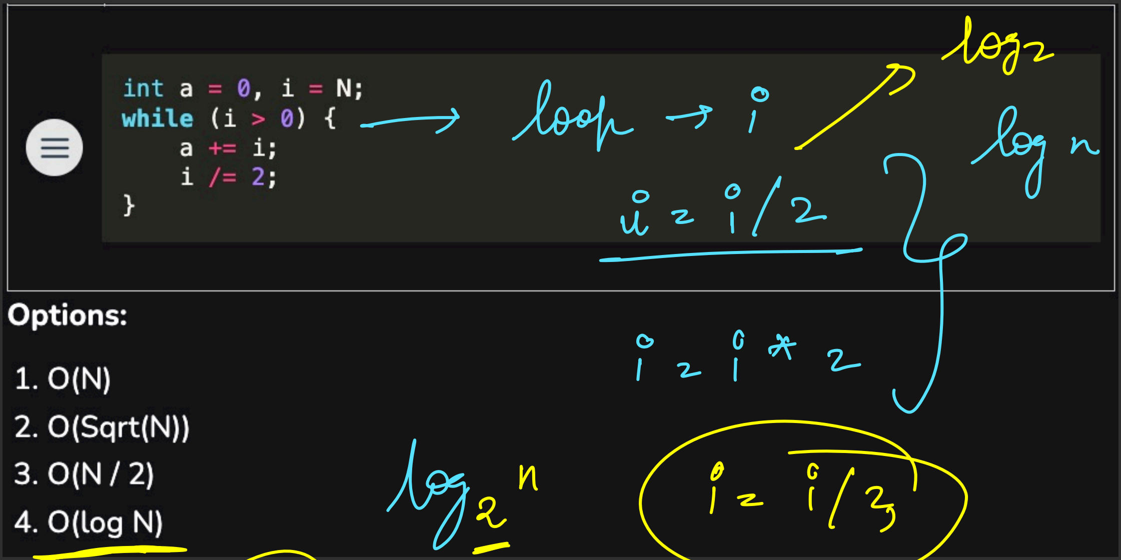
$\frac{N}{2} * \log n$

$\cancel{\log 5}$

$\log (n)$

constant

$N \log N$



```
int a = 0, i = N;
while (i > 0) {
    a += i;
    i /= 2;
}
```

$\longrightarrow$ loop $\rightarrow$ i $\nearrow$ $\log_2$

$\log n$

$i = i/2$

$i = i * 2$

Options:

1. O(N)
2. O(Sqrt(N))
3. O(N / 2)
4. O(log N)

$\log_2 n$

↳ $\log n$

$i = i/3$

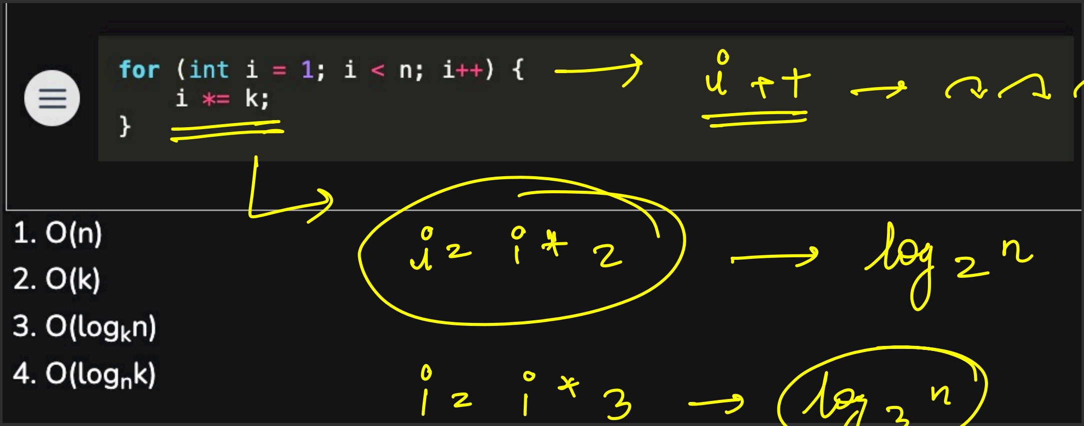
$>= n$

↳ $\log_3 n$

$3^k \sim n$

$\log_3 3^k \sim \log_3 n$

$k \sim \log_3 n$



```
for (int i = 1; i < n; i++) {
    i *= k;
}
```

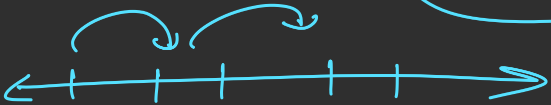
1. O(n)
2. O(k)
3. O($\log_k$n)
4. O($\log_n$k)

→ i++ →

$i = i * 2 \rightarrow \log_2 n$

$i = i * 3 \rightarrow \log_3 n$

$i = i * k \rightarrow \log_k n$



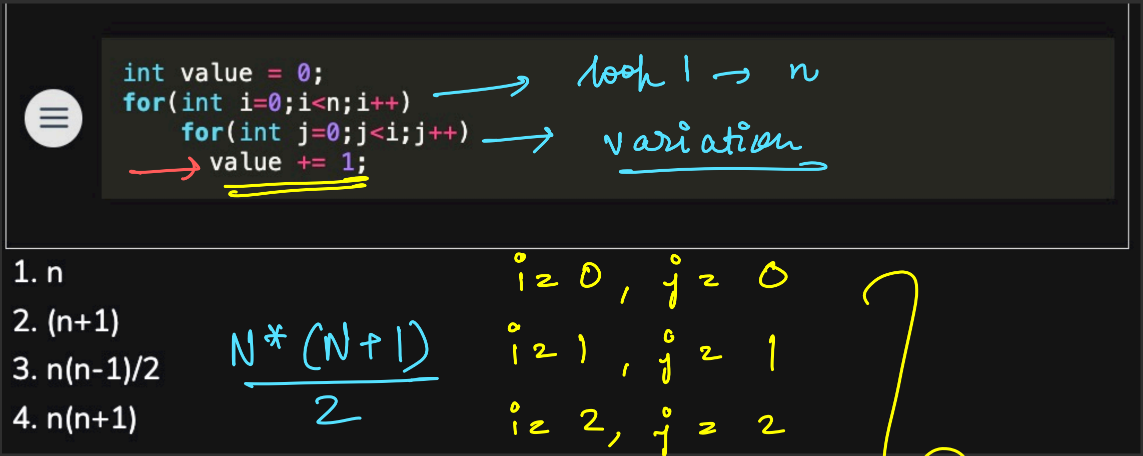
i++

↳ linear

Exponential grow

↓

1, 2, 4, 8, 16, 32



```
int value = 0;
for(int i=0;i<n;i++)
    for(int j=0;j<i;j++)
        value += 1;
```

loop 1 → n

variation

1. n
2. (n+1)
3. n(n-1)/2
4. n(n+1)

$$\frac{N*(N+1)}{2}$$

i = 0 , j = 0

i = 1 , j = 1

i = 2, j = 2

j = 3

j = 4

j = 5

n-1

$O(N^2)$

$N \sim (N-1)$

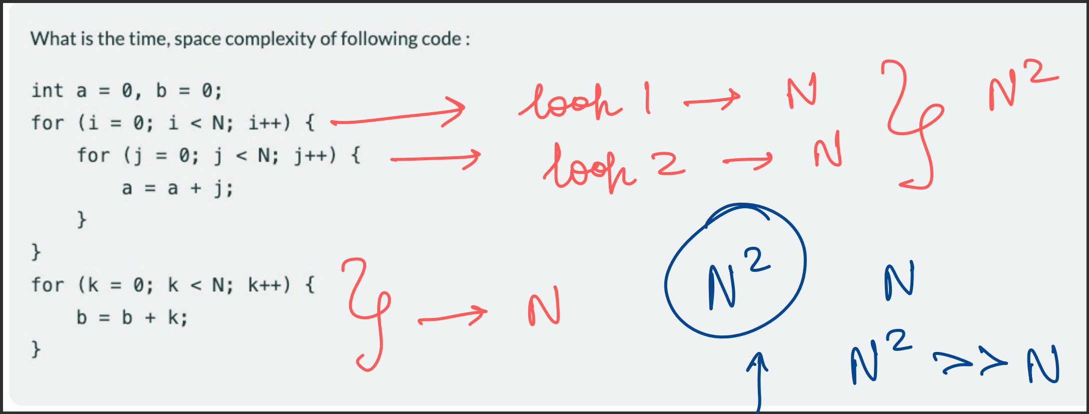
3rd ← $$\frac{(N-1)*N}{2}$$

$$\frac{N^2}{\cancel{2}} - \frac{N}{\cancel{2}}$$

$\boxed{2}$

N^2

$N^2 >> N$



What is the time, space complexity of following code :

```
int a = 0, b = 0;
for (i = 0; i < N; i++) {
    for (j = 0; j < N; j++) {
        a = a + j;
    }
}
for (k = 0; k < N; k++) {
    b = b + k;
}
```

loop 1 → N

loop 2 → N

N^2

→ N

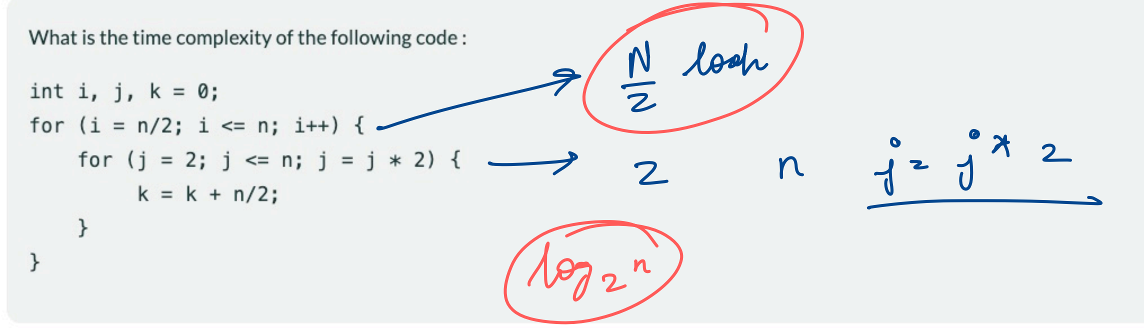
$\boxed{N^2}$

N

$N^2 >> N$

$\underline{\underline{N^2 >>> N}}$

$\underline{\underline{O(N^2)}}$



What is the time complexity of the following code :

```
int i, j, k = 0;
for (i = n/2; i <= n; i++) {
    for (j = 2; j <= n; j = j * 2) {
          k = k + n/2;
    }
}
```

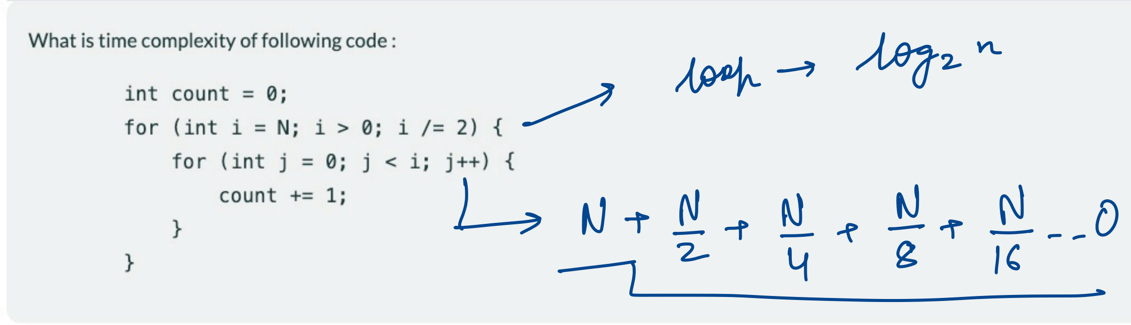
$\frac{N}{2}$ logn

2   n $j = j * 2$

$\log_2 n$

$\frac{N}{2} * \log_2 n$

$N * \log_2 n$ → Answer



What is time complexity of following code :

```
int count = 0;
for (int i = N; i > 0; i /= 2) {
    for (int j = 0; j < i; j++) {
        count += 1;
    }
}
```

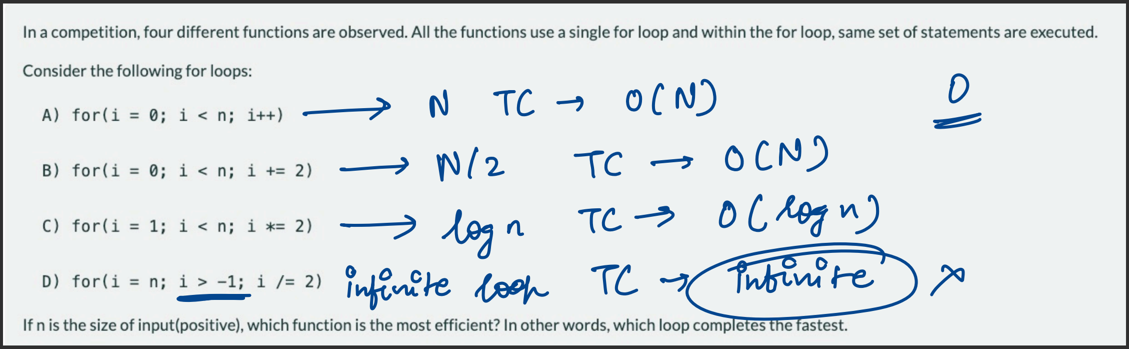
loop → $\log_2 n$

$N + \frac{N}{2} + \frac{N}{4} + \frac{N}{8} + \frac{N}{16} -- 0$

$N\left(1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} --- \right)$

$N * \left(1 + \frac{1}{2} + \frac{1}{4} --- \right)$

→ $\boxed{N * \log N}$ ↳ $\boxed{\log_2 n}$



In a competition, four different functions are observed. All the functions use a single for loop and within the for loop, same set of statements are executed.

Consider the following for loops:

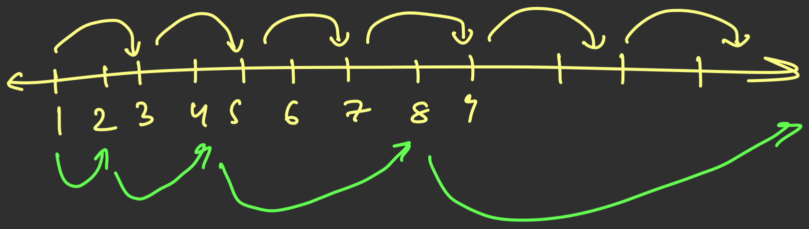
A) `for(i = 0; i < n; i++)` → N TC → $O(N)$ $\underline{\underline{0}}$

B) `for(i = 0; i < n; i += 2)` → N/2 TC → $O(N)$

C) `for(i = 1; i < n; i *= 2)` → log n TC → $O(\log n)$

D) `for(i = n; i > -1; i /= 2)` infinite loop TC → Infinite ✗

If n is the size of input(positive), which function is the most efficient? In other words, which loop completes the fastest.



$5 \to 2 \to 1 \to \underline{\underline{0}}$

$O(1) < O(\log N) < O(\text{sqrt}\, N) < O(N)$

$< O(N \log N) < O(N^2) < O(N^3)$

$< O(2^n) < O(3^n)$

$< O(N^n)$

↓

Recursion

i = 1 ; i < 5*N + 15 ; → x

x

└→ N

O(N)

infinite loop

zero

```
int a = 0;
for( int i = 0; i < N; i = i * 2)
{
    a = a + 5;
}
```

~~log n~~

i = i * 2 → 0