Bitmasking

Quick Recap – Bitwise operators

Bitwise ① AND

output

5 & 3 = 1

7 & 3 = 3

```
  000101
& 000011
--------
  000001
```

```
  000111
  000011
--------
  000011
```

Bitwise ② OR

5 | 3 = 7

5 | 8 = 13

```
  000101
| 000011
--------
  000111
```

```
  000101
  001000
--------
  001101
  8 4 2 1
```

③ NOT

(Unary op)

~5 = x

```
~ 00000101
= 11111010
```

Flip all bits

4 XOR (exclusive OR)

0 ^ 0 = 0
1 ^ 0 = 1
0 ^ 1 = 1
1 ^ 1 = 0

5 ^ 7 = 2

9 ^ 9 = 0

~~9~~ ^ 8 ^ ~~9~~ = 8

```
  0 0 0 1 0 1
^ 0 0 0 1 1 1
-------------
  0 0 0 0 1 0
```

5 Right Shift Operator >>

6 Left Shift <<, <<<

Java

>>

Filler element

Signed Right Shift

Fill 0 if MSB 0

Fill 1 if MSB 1

32nd bit (sign bit)

0 0 0 0 0 1 0 1 0

x x

0 0 0 0 0 0 1 0

N = 10

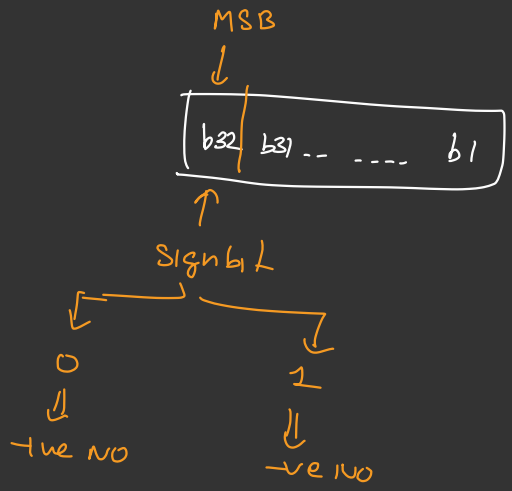
N = N >> 2

>>>

[never used]

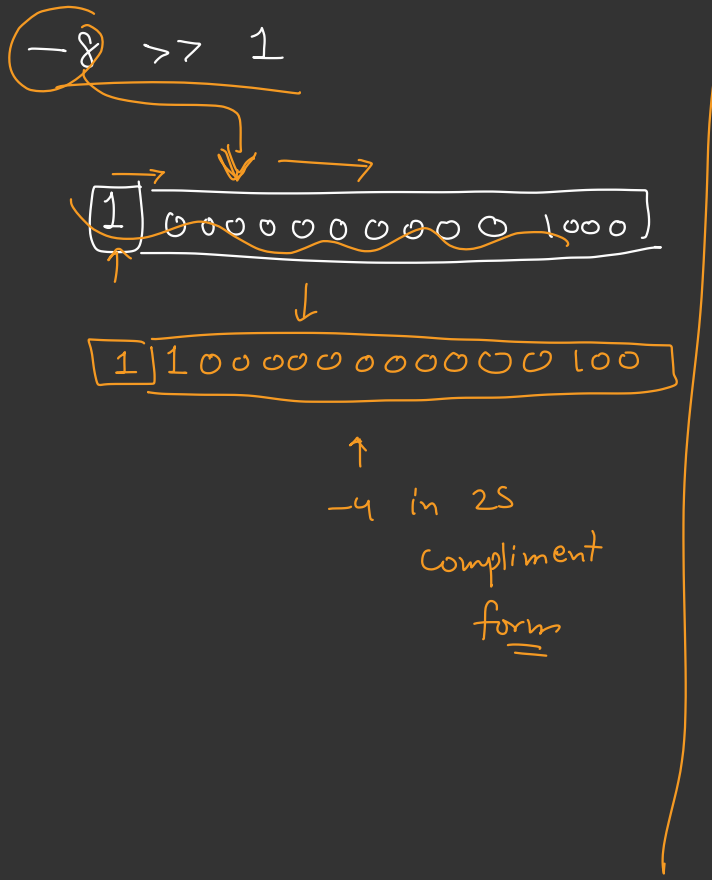
always fill 0

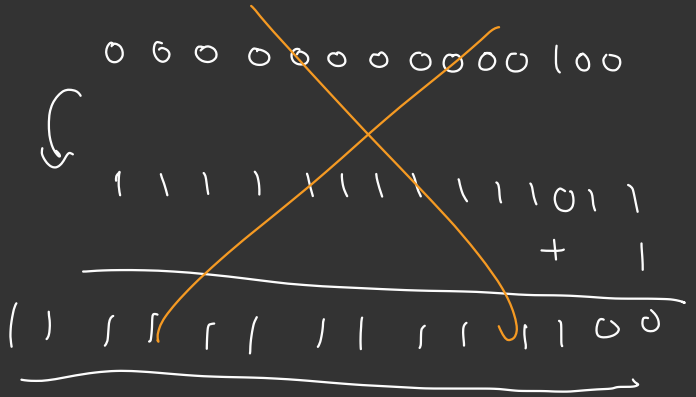
N >>> 3

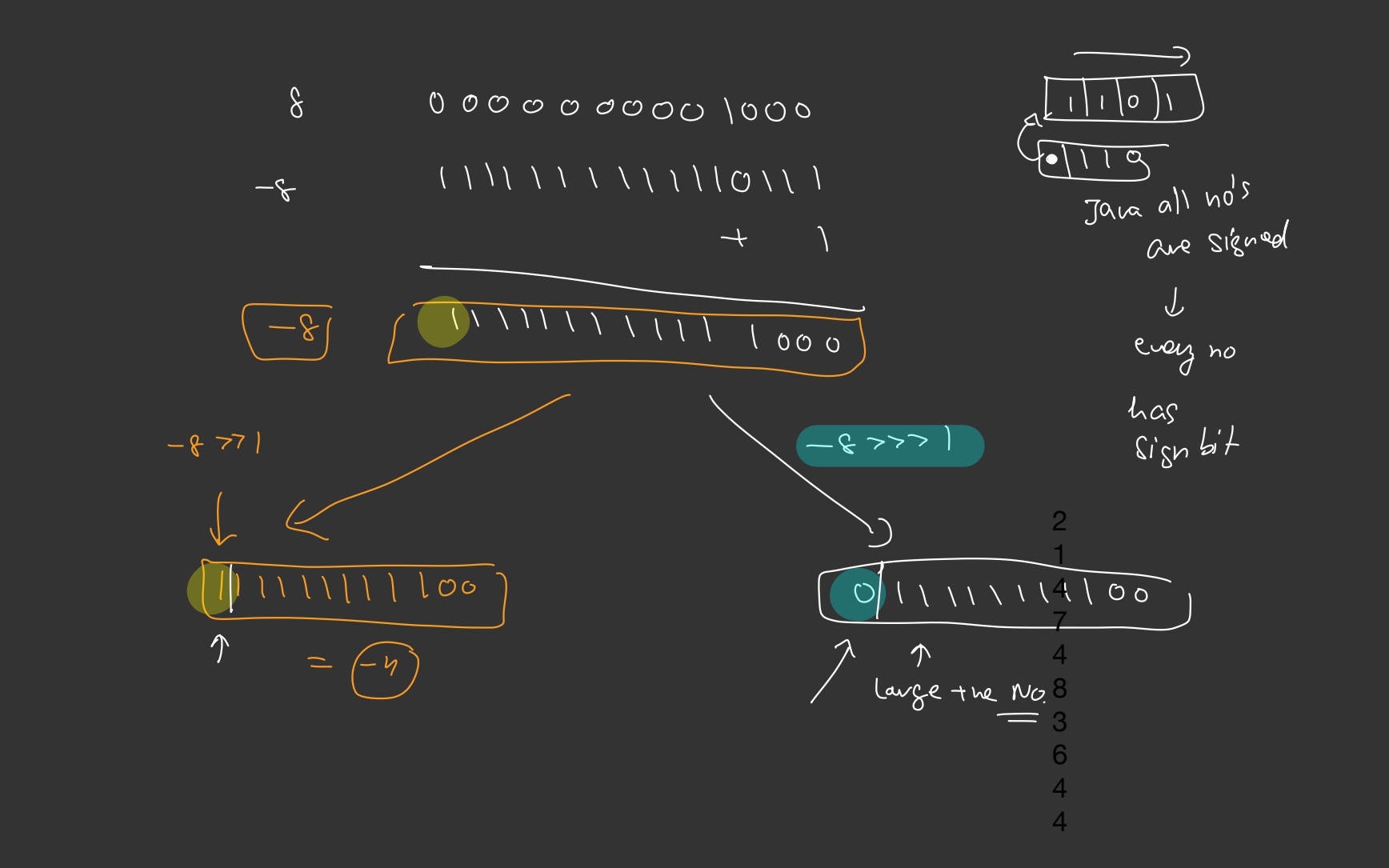


$-8 >> 1$

$= \boxed{-4}$


-8 >> 1
1 0000000000 1000
1 1000000000000 100
-4 in 2S Compliment form
-8 >>> 1
1 0000000 1000
30
0 1000000001 00
2^30 + 4




8
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
-8
1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1
+ 1
-8
1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
-8 >> 1
-8 >>> 1
1 1 1 1 1 1 1 1 1 0 0
= -4
0 1 1 1 1 1 1 1 1 0 0
large +ve no.
2147483640
Java all no's are signed
every no has sign bit

Recap

10 → 1010

10 >> 1 = 5 → 101

Right

a >> 1 → $\frac{a}{2}$

a >> 2 → $\frac{a}{2^2}$

10 >> 2

= 10 (10 crossed out)

= (2)

$\frac{10}{2^2}$ = (2)

a >> b → $\frac{a}{2^b}$ ✓

Left Shift

5 << 1 = 1010 = (10)

5 << 2 = 10100 = (20)

a << 2 = a * 2^2

a << b = a * 2^b ✓

PROBLEM

Unique No - III

Given an array containing 3N + 1 no's every element repeats twice except 1 unique NO. find unique no.

Ex - 6, 5, 8, 7, 7, 8, 8, 5, 2, 5, 6, 6, 7

1. Brute Force $O(N^2)$ + $O(1)$ space
2. Hashmap $O(N)$ + $O(N)$ space
   elem - cnt
3. Bitmasking $O(N)$ + $O(1)$ space

6, 5, 8, 7, 7, 8, 8, 5, 2, 5, 6, 6, 7

Unique → 8 ^ 7 ^ 2 ^ 6 ^ 5 = Rand No Not Helpful

6, 5, 8, 7, 7, 8, 8, 5, 2, 5, 6, 6, 7

	b4	b3	b2	b1
6	0	1	1	0
5	0	1	0	1
8	1	0	0	0
7	0	1	1	1
7	0	1	1	1

6 → 1 1 0
6 → 1 1 0
6 → 1 0 0

7
7
7

3, 3, 0
1, 1, 1
1, 1, 1
1, 1, 1
3, 3, 3

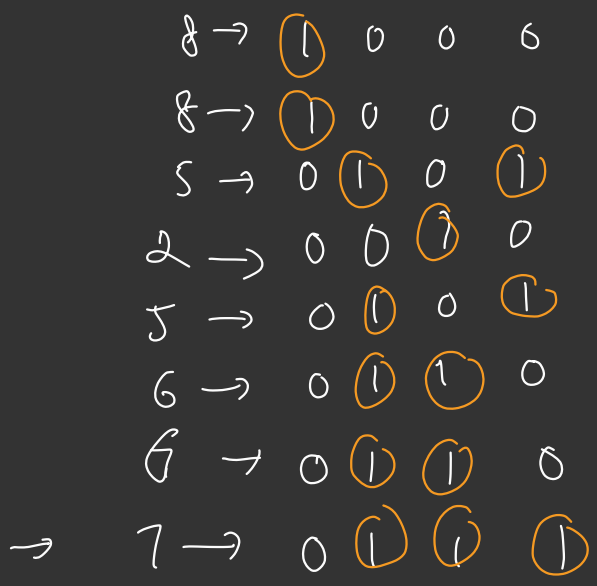
2 → 0, 1, 0

0, 0, 0 %3

6, 7, 3 %3

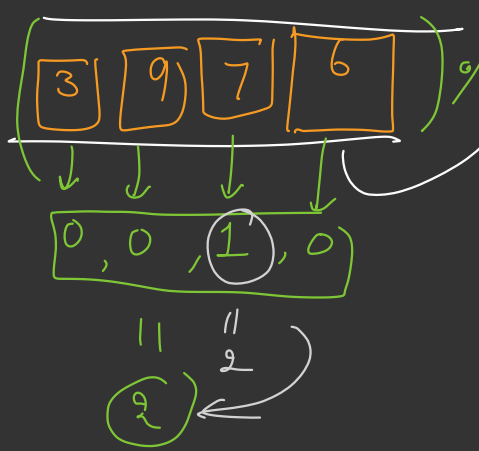
0, 1, 0

Biz by Unique No



No' 3rice → 3K bit at given once → 1 bit or 0 bit

Claim



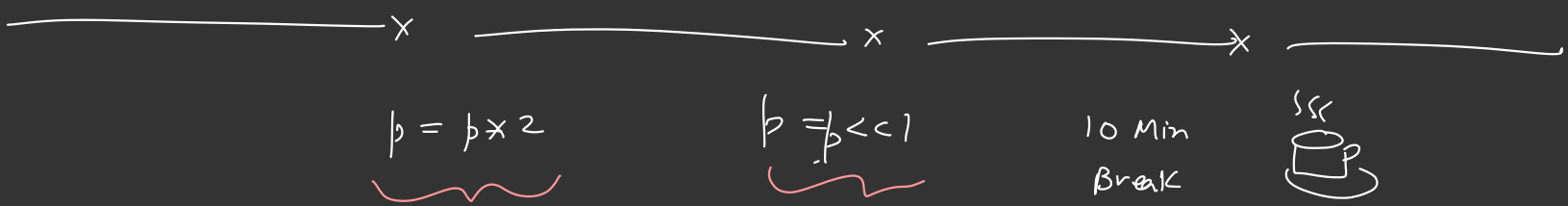
sum → (3K+1 or 3K) %3 = 1 , 0

Cancel out bit repeating thrice

$5N+1$

$7N+1$

Yes



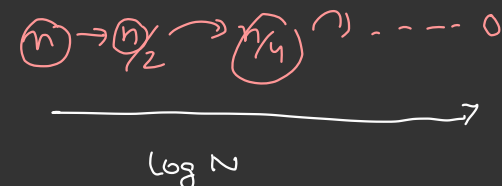
Finding Power a^n

$a=5, n=3$ $\quad 5\times5\times5 = 125$

① **Loop** multiply a, n times $\quad O(N)$, $O(1)$ space

② **Recursion**

$$a^n \begin{cases} \rightarrow (a^{n/2})^2 \\ \rightarrow a(a^{n/2})^2 \end{cases}$$



$f(a,n)$

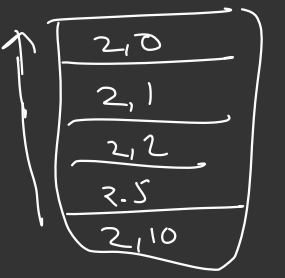
↓

$f(a,n/2)$

$O(\log N)$, $O(\log N)$

↑

Call stack

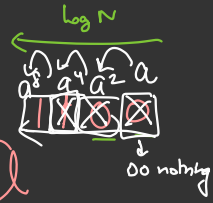


(3) Bitmasking

O(Log N)

O(1) space

a^{12} = a^{n}

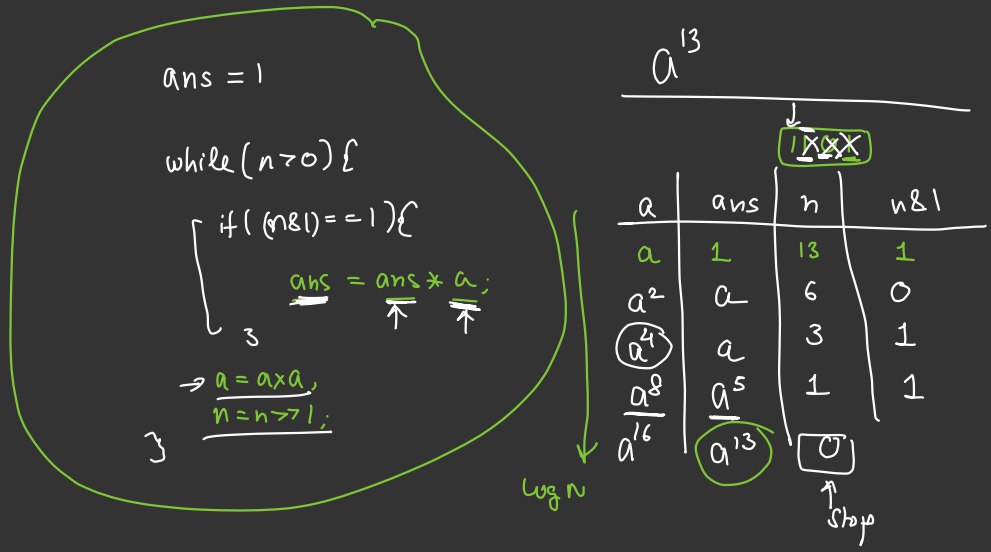


ans = $1 \times a^4 \times a^8$

$= a^{12}$

$= a^{1000} \cdot a^{0100} \cdot$

$= a^8 \cdot a^4 = a^{8+4} = a^{12}$


ans = 1
while (n > 0) {
if ((n&1) == 1) {
ans = ans * a;
}
a = a x a;
n = n >> 1;
}
Log n
a^13
1 X X X
a | ans | n | n&1
a | 1 | 13 | 1
a^2 | a | 6 | 0
a^4 | a | 3 | 1
a^8 | a^5 | 1 | 1
a^16 | a^13 | 0
Stop

⇒ Subsets using Bitmasking [Application]

⇒ Unique No - 2 [Intermediate Batch]

↳ 2N + 2 where every no coming twice except 2 unique no's.

→ 7, 6, 4, 5, 4, 5, 7, 8

Goal → 6, 8

Brute force → $O(N^2)$, $O(1)$ Space.

Hashmap → $O(N)$, $O(N)$ Space

Bitmasking →

① XOR = $7 \wedge 6 \wedge 4 \wedge 5 \wedge 4 \wedge 5 \wedge 7 \wedge 8$

$= 6 \wedge 8 \wedge 6$

$= 14 \neq 8$

A → 0 1 1 0

B → 1 0 0 0

XOR can't be zero.

→ atleast 1 set bit

3 2 1 0

$1 \wedge 0 = 1$

$1 \wedge 1 = 0$

(2) Pick any pos having a set bit
↳ is present in only one of the no's

p = 1

(3) filtering

p = 1

1(1)1 1(1)0 1(0)0 1(0)1 1(0)0 1(0)1 1(0)1 1(0)0

7 6 4 5 4 5 7 8

× × × × ✓ ×

A = [7, 6, 7] = 6

B = [4, 5, 4, 5, 8] = 8