-ve numbers ⟶ 2's complement

Decimal Numbers

[0 — 9]

Binary Numbers

[0 - 1]

- false → unset bit
- true → set bit

$(45)_{10} \rightarrow (?)_2$

2	45	1
2	22	0
2	11	1
2	5	1
2	2	0
2	1	1
	0	

$(101101)_2$

2	12	0
2	6	0
2	3	1
2	1	1
	0	

✓ $(12)_{10}$ ↓ $(1100)_2$

$(1100)_2 \longrightarrow$

$2^3\ 2^2\ 2^1\ 2^0$

3 2 1 0

1 1 0 0

$0 * 2^0 = 0$

$0 * 2^1 = 0$

$1 * 2^2 = 4$

$1 * 2^3 = 8$

$8 + 4 = 12$ ✓

$2^3\ 2^2\ 2^1\ 2^0$

3 2 1 0

1 1 0 0

$2^3 + 2^2 = 8 + 4 = 12$

6 5 4 3 2 1 0

$(1\ 0\ 1\ 1\ 0\ 1\ 0)_2$

$2^6 + 2^4 + 2^3 + 2^1 = 64 + 16 + 8 + 2 = 90$ ✓

Bitwise Operators

AND → All 1's ⇒ 1 else 0.

OR → All 0's ⇒ 0 else 1.

A	B	A & B (AND)	A | B (OR)	A ^ B (XOR)
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

XOR → "same same puppy shame"

Odd 1's ⇒ 1 else 0

int → 4 Bytes → 32 bits

int a = 5 → 101

int b = 6 → 110

print (a & b) → 100 → 4 ✓

(2 1 0)

20 → 0 1 0 1 0 0

45 → 1 0 1 1 0 1

```
  0 1 0 1 0 0
OR 1 0 1 1 0 1
-------------
  1 1 1 1 0 1
  5 4 3 2 1 0
```

$2^5 + 2^4 + 2^3 + 2^2 + 2^0 = 32 + 16 + 8 + 4 + 1$

$= 61$ ✓

```
  0 1 0 1 0 0
XOR 1 0 1 1 0 1
-------------
  1 1 1 0 0 1
  5 4 3 2 1 0
```

→ $2^5 + 2^4 + 2^3 + 2^0 = 32 + 16 + 8 + 1 = 57$ ✓

Properties

Eg → A = 5 → $(101)_2$ 0 → $(000..0)_2$

1) A & 0 = 0

```
  1 0 1
& 0 0 0
-------
  0 0 0 → 0
```

2) A & A = A

```
  1 0 1
& 1 0 1
-------
  1 0 1 → A
```

3) A | 0 = A

```
  1 0 1
| 0 0 0
-------
  1 0 1 → A
```

4) A | A = A

```
  1 0 1
| 1 0 1
-------
  1 0 1 → A
```

5) A ^ 0 = A

```
  1 0 1
^ 0 0 0
-------
  1 0 1 → A
```

6) A ^ A = 0

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

7) Odd Even check

n % 2 → 1 ⇒ odd

→ 0 ⇒ even

Odd → 5 = 101
9 = 1001
11 = 1011
last bit = 1

Even → 6 = 110
10 = 1010
12 = 1100
last bit = 0

A & 1 → last bit of A

A = 9 → 1001
1 → & 0001
0001 → 1

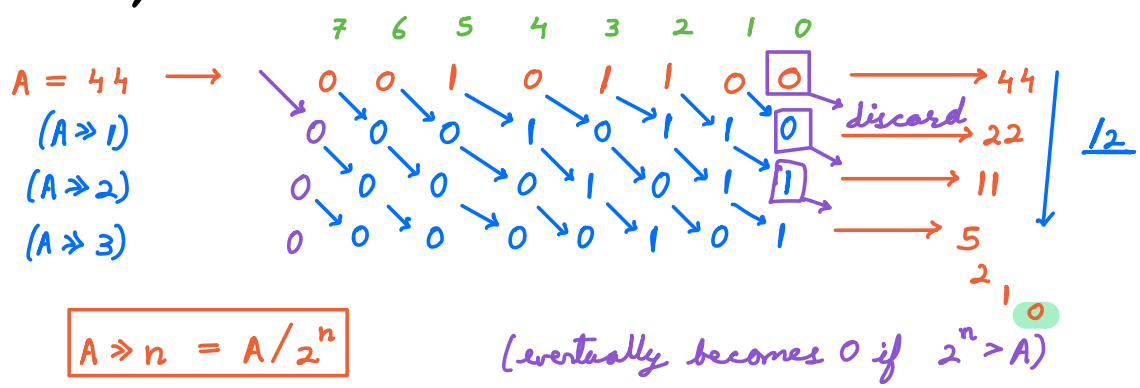
12 → 1100
1 → 0001
0000 → 0

A & 1 → 1 ⇒ odd
→ 0 ⇒ even

Not ! / ~ bitwise 1 → 0
0 → 1

Left Shift & Right Shift
≪ ≫

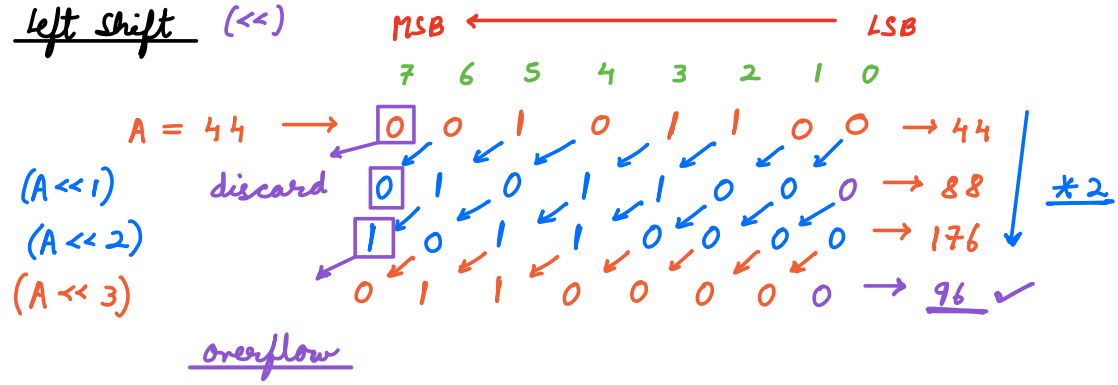
Right Shift (≫) 8 bit number 5/2 = 2



A ≫ n = A / 2^n

(eventually becomes 0 if 2^n > A)

Left Shift (≪)



overflow

$A \ll n = A * 2^n$

8 bits (+ve numbers)

7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1

$2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 255$ ✓

$A = 44$

$A \ll 3 \rightarrow 44 * 2^3$

$= 44 * 8 = 352$ (expected)

Q → check if i^{th} bit is set for a number N.

$N = 44$

5	4	3	2	1	0
1	0	1	1	0	0

$i = 2 \rightarrow$ Ans = true

$i = 4 \rightarrow$ Ans = false

A & 1 → last bit of A

$(A \gg i) \& 1$ → 1 → Ans = true; 0 → Ans = false

TC = O(1)

SC = O(1)

```
if (a+2 < 5) {...}
if ((N >> i) & 1 == 1) {...}
```
} ✓

$N = N \gg i$ → to change N.

$N = 44$

5	4	3	2	1	0
1	0	1	1	0	0

$i = 2$

$2^i = (1 \ll i) \rightarrow$

0	0	0	1	0	0

&

0	0	0	1	0	0

$N \& (1 \ll i)$ → $(1 \ll i) \Rightarrow$ Ans = true; 0 ⇒ Ans = false

---

10:40 PM

$a * 5$ ✓

$a + a + a + a + a$ ✗

$A \wedge 0 = A$

$A \wedge A = 0$

$2 \wedge 2 \wedge 3 = 0 \wedge 3 = 3$

$2 \wedge 3 \wedge 2 = 0 \wedge 3 = 3$

$2+2+3 = 2+3+2$

$2 \wedge 2 \wedge 3 = 2 \wedge 3 \wedge 2$

Associative Property

$2 \wedge 3 \wedge 4 \wedge 2 \wedge 4 = 2 \wedge 2 \wedge 4 \wedge 4 \wedge 3$

$= 0 \wedge 0 \wedge 3 = 3$

$a \wedge b = b \wedge a$

$6 \wedge 5 \wedge 3 \wedge 1 \wedge 5 \wedge 1 \wedge 6 = 3$ ←

Q → Given an integer array where every element appears twice except for one element that appears once. Find that unique element.

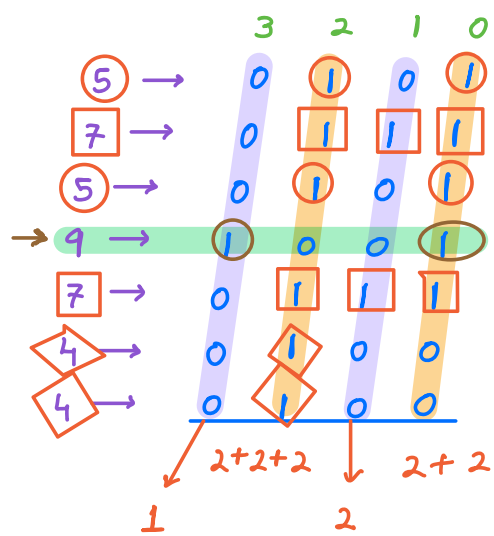
$A = [2 \quad 3 \quad 5 \quad 6 \quad 3 \quad 6 \quad 2] \rightarrow 5$

```
ans = 0
for i → 0 to (N-1)
    ans ^= A[i]
return ans
```

TC = O(N)

SC = O(1)

$A = [5 \quad 7 \quad 5 \quad 9 \quad 7 \quad 4 \quad 4]$



XOR → 1 0 0 1 → 9 → (unique element)

$1 \wedge 1 \wedge 1 \ldots \wedge 1$ → even → 0

odd → 1

unique element contribution

Q → Given an integer array where every element appears 3 times except for one element that appears once. Find that unique element.

A = [2 3 6 2 2 3 3] → 6

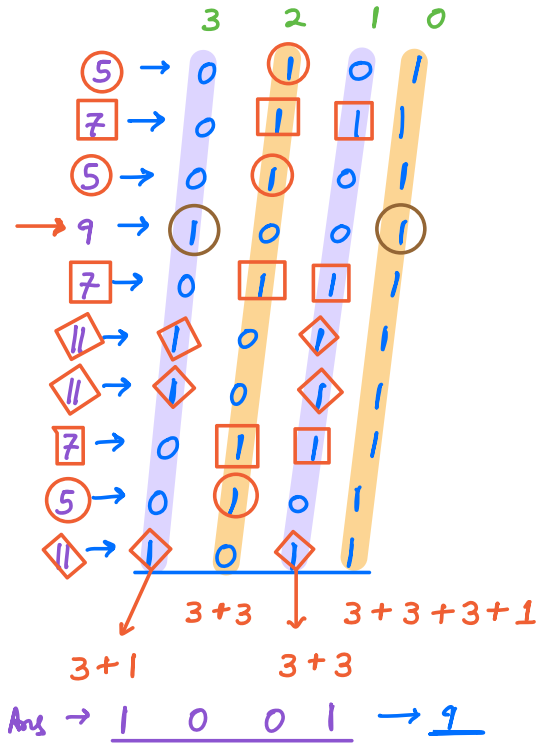
A = [5 7 5 9 7 11 11 7 5 11] → 9

XOR of all elements = ~~x~~ ^ ~~3~~ ^ 6 ^ ~~x~~ ^ 2 ^ ~~3~~ ^ 3 = 6 ^ 2 ^ 3 (not working)

Bruteforce → ∀A[i], travel & check if it is unique. → TC = O(N²)
SC = O(1)

Freq → ∀A[i], store frequency & Ans = A[i] with freq = 1.
TC = O(N) ✓
SC = O(N) ←

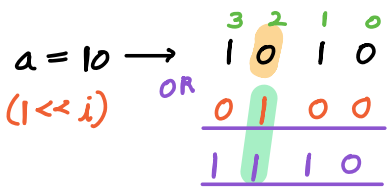
A = [5 7 5 9 7 11 11 7 5 11]



#1's
→ 3+3+3+... 3 → 3K
→ 3+3+3+... 3 + 1 → 3K+1
unique element
x % 3 == 1
#1's

---

ans = 0
for b → 0 to 31 // int
// count #1's at b^th position
cnt = 0
for i → 0 to (N-1)
cnt += (A[i] >> b) & 1




```
if (cnt%3 == 1)
    // set ith bit in ans
    ans |= (1<<b)
```

return ans

TC = O(32*N) ≈ O(N)

SC = O(1)

Q → Given an integer array where every element appears 4 times except for one element that appears once. Find that unique element.

→ Ans = XOR of all elements

Q → Given an integer array where every element appears 5 times except for one element that appears once. Find that unique element.

→ Manually like (3 times)

Q → Given an integer array where every element appears K times except for one element that appears once. Find that unique element.

- → K is even ⇒ Ans = XOR of all elements
- → K is odd ⇒ Manually like (3 times)
