Bit Manipulation

Decimal → Binary — 0's or 1's

Real World

1st Key Takeaway

- ↳ Efficient & fast
- ↳ Hardware
- ↳ Compression Algorithms
- ↳ Encryption

Bitwise

① Operators

- Left shift (<<)
- Right shift (>>)

2nd Key Takeaway

num = 4

4 >> 1

100 = 4
010 = 2

Division by 2

num = 4

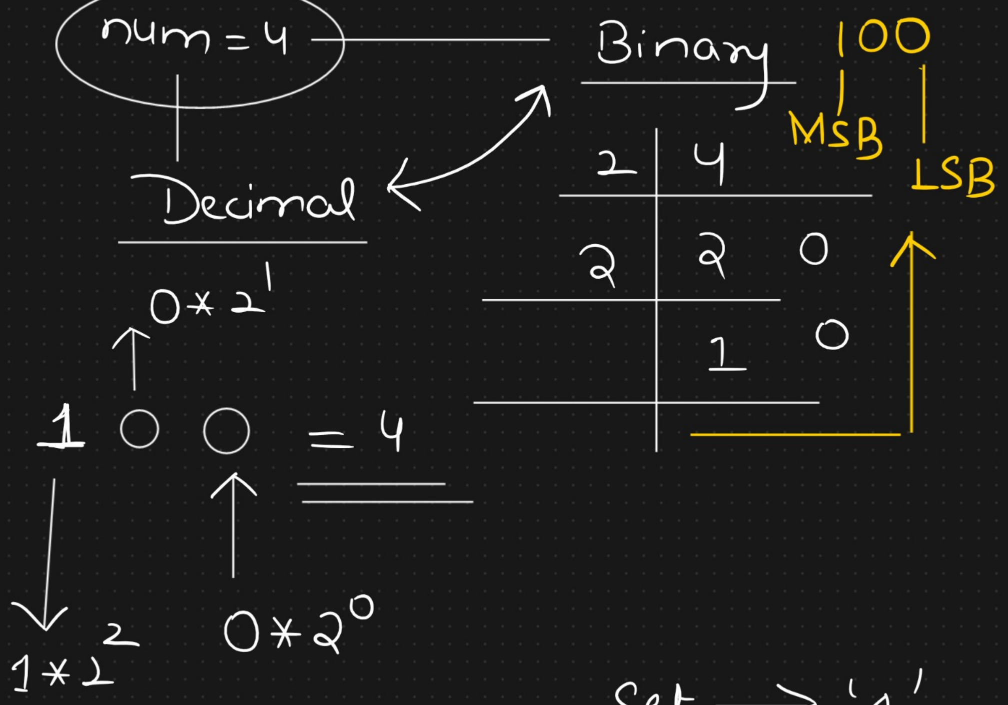
4 << 1

↳

0100 = 4
1000 = 8

10000 = 16

multiplication by 2



Set → '1'

3 KeyTakeaway

OR(|)

A	B	o/p
0	0	0
0	1	1
1	0	1
1	1	1

AND(&)

A	B	o/p
0	0	0
0	1	0
1	0	0
1	1	1

XOR(^)

A	B	o/p
0	0	0
0	1	1
1	0	1
1	1	0

$$\boxed{\begin{aligned} A \wedge A &= 0 \\ A \wedge 0 &= A \end{aligned}}$$

$$\begin{cases} 1 \wedge 0 = 1 \\ 4 \wedge 0 = 4 \\ 7 \wedge 0 = 7 \end{cases}$$

4th Key Takeaway

Interview Problems

①

0	1	2	3	4	5	6	7	8
2	4	2	4	5	6	6	5	4

$2 \wedge 4 \wedge 2 \wedge 4 \wedge 5 \wedge 6 \wedge 6 \wedge 5 \wedge \underline{4}$

$0 \wedge 4 = 4$

① Duplicate element

$2 \wedge 2 = 0$

②

2	2
$\underline{4}$	3
5	2
6	2
	9

odd num of times

$5 \wedge 5 = 0$

Result = 4

$6 \wedge 6 = 0$

$4 \wedge 4 \wedge 4$

$\underline{0} \wedge \underline{4} = 4$

XOR

```
getOddfreqElem(nums)
    for( int num: nums) {
        XOR = XOR ^ num;
    }
return XOR;
```

time complexity → $O(n)$

Space complexity → $O(1)$

(2)

num = 16 — PowerOfTwo → True

2^4 → True

AND

num = 15 — false

16 — 10000

AND

15 — 01111

$00000 \Rightarrow 0$

$$\underbrace{(num \,\&\, num-1) == 0}$$

num → Power of Two

③ num = 15   1 1 1 1

num = 4 → 100

O/p = 1   (1's)

count of the Result

Shift Operators

number of set bits = 4

While(num > 0) {

count = ~~0~~ ~~1~~ ~~2~~ ~~3~~ 4

num = 15

1 1 1 1
0 0 0 1

count += 0 0 0 1

① num & 1 count

② num >> 1 (/2)

0 1 1 1
0 0 0 1

0 0 0 1

0 0 0 0

~~0 0 0 1~~

0 0 0 1

0 0 0 1

4

0 0 1 1
0 0 0 1

0 0 0 1