- Truth Table
- Basic AND XOR OR Properties
- Single Number :
- Left Shift & Right Shift
- Power of left shift operator (set, flip, unset, check)
- Single Number 3 (Every element 2 times and 2 unique elements)

Truth table —

a = 10
a = 789

$(57)_{10} \longrightarrow (111001)_2$

Bit wise operators

1 bit is 0, ans will be 0

any bit is 1 ans is 1

a	b	a & b	OR a | b	XOR a ^ b
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

same same

puppy shame

& (AND)
| (OR)
^ (XOR)

$\longrightarrow$ 2 bits

NOT Operator

~ (tilda)

[! conditions]

✓

a	~a
0	1
1	0

$\Rightarrow$ $57 = (111001)_2$

$\sim 57 = (000110)_2$

Even & Odd number

Odd :

57 : a = 111001

15 : 1111

21 : 10101

```
int a = 57;

if (a & 1 == 1)
    Odd

elsif (a & 1 == 0) {
    Even
}
```

```
  111001
& 000001
  ------
  000001
```

a = 56

```
  111000
& 000001
  ------
  000000
```

```
21 = 10101
   & 00000
     -----
     00000
```

```
23 = 10111
   & 10111
     -----
     10111
```

23

a & 0 = 0

a & a = a

OR

$a \mid 0 = a$

$a \mid a = a$

$2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$

$48 = 110000$

$\text{OR}\ 000000$

$110000 \rightarrow 48$

$49 = 110001$

$\text{OR}\ 110001$

110001

^

$a \wedge 0 = a$

$a \wedge a = 0$

$57 = 111001$

$\wedge\ 000000$

111001

$57 \quad 111001$

$\wedge 57 \quad = \wedge\ 111001$

000000

$= 1 \wedge 2 \wedge 3 \wedge 3 \wedge 2 \wedge 1$

001

$\wedge\ 010$

011

$= (3 \wedge 3) \wedge 3 \wedge 2 \wedge 1$

0

$$= (\underline{0 \wedge 3}) \wedge 2 \wedge 1$$

$$= (3 \wedge 2) \wedge 1$$

$$\begin{array}{r} 011 \\ 010 \\ \hline 001 \\ \hline \end{array}$$

$$= (1 \wedge 1)$$

$$= 0$$

$$1 \wedge 2 \wedge 3 \wedge 3 \wedge 2 \wedge 1 = 0$$

L ← R

$$1 \quad 2 \wedge 3 \wedge \underbrace{3 \wedge 3}_{0}$$

$$1 \wedge 2 \wedge \underbrace{3 \wedge 0}$$

$$1 \wedge \underbrace{2 \wedge 3}$$

$$\underline{1 \wedge 1} = 0$$

$1 \wedge 3 \wedge 2 \wedge 2 \wedge 1 \wedge 3$

$$\begin{array}{r} 001 \\ 011 \\ \hline 010 \\ \hline \end{array} = 2$$

$(2 \wedge 2) \wedge 2 \wedge 1 \wedge 3$

0

$(0 \wedge 2) \wedge 1 \wedge 3$

$$\begin{array}{r} 010 \\ 001 \\ \hline 011 \\ \hline \end{array} = 3$$

$2 \wedge 1 \wedge 3$

$3 \wedge 3 = 0$

You can take elements, $\wedge$ will always be same —

$a \wedge b \wedge c = c \wedge b \wedge a$

$c \wedge a \wedge b = b \wedge c \wedge a$

Negative no's ✓

XOR = subtraction No

⇒ any order → $a \wedge b = b \wedge a$

$a \wedge b == b \wedge a$

$\underbrace{57 \wedge 21}$ $\qquad$ $21 \wedge 57$

$$\begin{array}{r} 111001 \\ \wedge\ 010101 \\ \hline 101100 \end{array} \qquad \begin{array}{r} 010101 \\ \wedge\ 111001 \\ \hline 101100 \end{array}$$

$1 \wedge 0 == 0 \wedge 1$

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

⇒

$16 \wedge 17 \wedge 21 \wedge 57$

16	010000
17	010001
21	010101
57	111001
	----01

$\cancel{0} \wedge 1 \wedge 1 \wedge 1$

$0 \wedge 1 = 1$ ✓

$17 \wedge 16 \wedge 57 \wedge 21$

17	010001
16	010000
57	111001
21	010101
	01

$\underbrace{\cancel{1} \wedge 0} \wedge 1 \wedge 1$

$1 \wedge 1 \wedge 1$

$0 \wedge 1 = 1$

$$1 \wedge 0 \wedge 1 \wedge 0 = 0$$

$$1 \wedge 1 \wedge 0 \wedge 0 = 0$$

$$0 \wedge 1 \wedge 0 \wedge 1 = 0$$

$$\Rightarrow \quad 0 \wedge 0 \quad 0 \quad 0$$

$$0 \wedge 0$$

$$0 \wedge 0$$

$$0$$

---

any order of number $\longrightarrow$ xor value be same

- and
- or

$\longrightarrow$ true ✓

Q1 $a \wedge b \wedge a \wedge d \wedge b$

$$\underbrace{\cancel{(a \wedge a)}}_{0} \wedge b \wedge b \wedge d$$

$$\underbrace{\cancel{(0 \wedge b)}}_{b} \wedge b \wedge d$$

$$\cancel{(b \wedge b)} \wedge d$$

$$(0 \wedge d) = d \quad \underline{\text{ans}}$$

Q2 $1 \wedge 3 \wedge 5 \wedge 3 \wedge 2 \wedge 1 \wedge 5$

$$\overset{0}{\cancel{(1 \wedge 1)}} \wedge (3 \wedge 3) \wedge (5 \wedge 5) \wedge 2$$

$0 \wedge a = a$

$$\underbrace{\cancel{0 \wedge 3}}_{3} \wedge 3 \wedge 5 \wedge 5 \wedge 2$$

$a \wedge a = 0$

$$\overset{0}{\cancel{(3 \wedge 3)}} \wedge 5 \wedge 5 \wedge 2$$

$$\underbrace{\cancel{(0 \wedge 5)}}_{5} \wedge 5 \wedge 2$$

$$\underbrace{(5 \wedge 5)}_{0} \wedge 2$$

$$0 \wedge 2 = \underline{2} \text{ ans}$$

Q3 $\Rightarrow$ $120 \wedge 5 \wedge 6 \wedge 6 \wedge 120 \wedge 5$

$$\underbrace{120 \wedge 120}_{0} \wedge 5 \wedge 5 \wedge 6 \wedge 6$$

$$\underbrace{(0 \wedge 5)}_{5} \wedge 5 \wedge 6 \wedge 6$$

$$\underbrace{5 \wedge 5}_{0} \wedge 6 \wedge 6$$

$$\underbrace{0 \wedge 6}_{} \wedge 6$$

$$\underline{6 \wedge 6} = 0 \text{ ans}$$

Q. find the single number ⟶

every number is repeating twice, one element is unique

find that unique element →

57	66	79	66	81	57	81

$57 \wedge 66 \wedge 79 \wedge 66 \wedge 81 \wedge 57 \wedge 81$

$= \cancel{57 \wedge 57}^{0} \wedge 66 \wedge 66 \wedge 81 \wedge 81 \wedge 79$

$= \cancel{0 \wedge 66}^{66} \wedge 66 \wedge 81 \wedge 81 \wedge 79$

$= \cancel{66 \wedge 66} \wedge 81 \wedge 81 \wedge 79$

$= \cancel{0 \wedge 81}^{81} \wedge 81 \wedge 79$

$= 0 \cancel{81 \wedge 81} \wedge 79 \quad = 0 \wedge 79 \quad = 79$ ans

$a \wedge 0 = a$

xor of all no's

xor = 0

for(i=0 i<N i++){

xor = xor ^ a[i]

}

xor

TC = O(N)

SC = O(1)

$a + 0 = a$

sum of all no's

sum = 0

for(i=0 i<N i++){

sum = sum + a[i]

}

sum

57 86 61 57 61

xor = 0

=> xor = 0 ^ 57 = 57

=> xor = 57 ^ 86

=> xor = (57 ^ 86) ^ 61

$\Rightarrow \quad XOR = (57 \wedge 86 \wedge 61) \wedge 57$

$\Rightarrow \quad XOR = (57 \wedge 86 \wedge 61 \wedge 57) \wedge 61$

$= 57 \wedge 86 \wedge 61 \wedge 57 \wedge 61$

$= 86 \wedge \cancel{(57 \wedge 57)}^{0} \wedge \cancel{61 \wedge 61}^{0}$

$= \underline{86}$

XOR

AND

OR

$+$

New

15 Left shift

5 Right shift

```java
// Use this editor to write, compile and run your Java co
public class Main {
    public static void main(String[] args) {
        int a = 5; // 101
        System.out.println(a << 0); // 101
        System.out.println(a << 1); // 1010
        System.out.println(a << 2); // 10100
        System.out.println(a << 3); // 101000
        System.out.println(a << 4); // 1010000
    }
}
```

Custom Output

[Success] Your code was executed successfully

```
5
10
20
40
80
```

Custom Input

32 Int $\underbrace{a}$ $= (101)_2 = (5)_{10}$

32 Int $\underbrace{a << 1}$ 1010

32 Int $\underbrace{a << 2}$ 10100

32 Int $\underbrace{a << 3}$ 101000

32 Int $\underbrace{a << 4}$ 1010000

8 bits example

a = 0 0 0 0 0 1 0 1 5

x2

a<<1 =	0	0	0	0	1	0	1	0	10
×2 a<<2 =	0	0	0	1	0	1	0	0	20
×2 a<<3 =	0	0	1	0	1	0	0	0	40
×2 a<<4 =	0	1	0	1	0	0	0	0	80
×2 a<<5 =	1	0	1	0	0	0	0	0	160
a<<6 =	0	1	0	0	0	0	0	0	320

320?

8 bits

all +ve

0 — 255

320 ✗

-ve — +ve

-2^7 to $2^7 - 1$

-128 to 127

320 ✗

Overflow

Before overflow

$\times 2$

$a << i = a * 2^i$

$a = 5 = 5 \times 2^0$

$a << 1 = 10 = 5 \times 2^1$

$a << 2 = 20 = 5 \times 2^2$

$a << 3 = 40 = 5 \times 2^3$

$a << 4 = 80 = 5 \times 2^4$

$\vdots$

```java
// Use this editor to write, compile and run your Java co
public class Main {
    public static void main(String[] args) {
        int a = 5; // 101
        System.out.println(a << 0); // 101
        System.out.println(a << 1); // 1010
        System.out.println(a << 2); // 10100
        System.out.println(a << 3); // 101000
        System.out.println(a << 4); // 1010000
    }
}
```

Custom Output

[Success] Your code was executed successfully

```
5
10
20
40
80
```

Custom Input

Right Shift

$a = 20$		$10100 \longrightarrow$	$\frac{20}{1}$
$a >> 1$	$\frac{a}{2^1}$	1010	$\frac{20}{2}$
$a >> 2$	$\frac{a}{2^2}$	101	$\frac{20}{4}$
$a >> 3$	$\frac{a}{2^3}$	10	$\frac{20}{8}$

$\Rightarrow \quad a >> i \quad \cdots\rangle \quad \frac{a}{2^i}$

$a = 7$

$a << 2 \qquad 111\,00$

$a * 2^2$

$7 * 2^2 = 28$

$\boxed{1 << 3}$

$1000 = 8$

4 applications of left shift operator →

Set ith bit

	i	5	4	3	2	1	0
57 =		1	1	1	0	0	1

set i = 2 bit
make it 1

1	1	1	1	0	1

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

set i = 3

1	1	1	0	0	1

	i	5	4	3	2	1	0
a =		1	1	1	0	0	1
OR		0	0	0	1	0	0
		1	1	1	1	0	1

set i=2

a | 1<<2

a | 1<<i → ith bit

set ith bit → a | 1<<i

a = 57   111001 ✓

1<<2   000100

Toggle ith bit

	5	4	3	2	1	0
a = 48	1	1	0	0	0	0
^ (i = 2)	0	0	0	1	0	0
	1	1	0	1	0	0

$$\begin{array}{r} 1 \\ \wedge\ 1 \\ \hline 0 \end{array} \quad 1 << 2$$

Toggle ith bit = $a \wedge (1 << i)$

⇒ Check ith bit is 1 or 0

a = 45

i = 2

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

$\boxed{1 << 2}$

if (a & (1 << i) > 0)

ith bit is 1

else

ith bit is 0

$a = 57$

$i = 4$

1 or 0

$1 << i$

$$
\begin{array}{rcccccc}
 & 1 & 1 & 1 & 0 & 0 & 1 \\
\& & 0 & 1 & 0 & 0 & 0 & 0 \\
\hline
 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}
\quad > 0
$$

Unset ith bit $\longrightarrow$

$a = 57$

$i = 4$

$$
\begin{array}{rcccccc}
 & 5 & 4 & 3 & 2 & 1 & 0 \\
 & 1 & 1 & 1 & 0 & 0 & 1 \\
 & 0 & 1 & 0 & 0 & 0 & 0 \\
\sim \& & 1 & 0 & 1 & 1 & 1 & 1 \\
\hline
 & 1 & 0 & 1 & 0 & 0 & 1
\end{array}
$$

ith bit

$1 \& 1 = 1$

$0 \& 1 = 0$

$\sim(1 << i)$

$$a \ \& \ (\sim(1 << i))$$

$a = 57$

$i = 2$

$a \wedge (1 << i)$

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

✓ Check ith bit

→ ith bit is 0

ignore, unset

→ ith bit is 1

$1 \longrightarrow 0$

Toggles

Q1 ⇒ every no repeats twice, one element is unique

find that unique

XOR of all numbers

Q2 **Problem Statement**

Given an integer array, all the elements will occur twice except two. Find those two elements.

Input: [4, 5, 4, 1, 6, 6, 5, 2]

Output: 1, 2

1 and 2 are unique

a b

a ^ b

Xor of all nos = 1 ^ 2

4 ^ 5 ^ 4 ^ 1 ^ 6 ^ 6 ^ 5 ^ 2

~~4 ^ 4~~ ^ ~~5 ^ 5~~ ^ ~~6 ^ 6~~ ^ 1 ^ 2 = 1 ^ 2

1 ^ 2 = 3

001

010

011

2 1 0

3

any one i → xor value = 1

i = 1

a = 0 ^ 4 ^ 5 ^ 4 ^ 1 ^ 5 = 1

unset

b = 0 ^ 6 ^ 6 ^ 2 = 2

set

4 5 4 1 6 6 5 2

001 110 101 010

100 101 100 — 110

7 6 5 1 7 5 3 6

$\Rightarrow$ 7 ^ 6 ^ 5 ^ 1 ^ 7 ^ 5 ^ 3 ^ 6

$\cancel{7 \wedge 7} \wedge \cancel{6 \wedge 6} \wedge \cancel{5 \wedge 5} \wedge 1 \wedge 3 = \underline{1 \wedge 3}$

$= 2$

$$\begin{array}{r} 001 \\ \wedge\ 011 \\ \hline 010 \\ \hline \end{array} = 2$$

$i = 1$

a = 0 ^ 5 ^ 1 ^ 5

unset

1

b = 0 ^ ~~7~~ ^ ~~6~~ ^ ~~7~~ ^ 3 ^ ~~6~~

set

3

7	6	5	1	7	5	3	6
111	110	101	001	111	101	011	110

4 4 2 7 3 7

=> 4 ^ 4 ^ 2 ^ 7 ^ 3 ^ 7

4 ^ 4 ^ 7 ^ 7 ^ 2 ^ 3 = 2 ^ 3 = 1

010 2

011 3

001 ← ✓

i = 0

a = 0 ^ 4 ^ 4 ^ 2

unset = 2

b = 0 ^ 7 ^ 3 ^ 7

set = 3

4	4	2	7	3	7
100	100	010	111	011	111

1)

```
xor = 0

for ( i = 0        i < N        i++) {
        xor = xor ^ a[i]
}
```

2) <u>xor</u> j = 0 <u>0 to 31</u>

```
while ( j < 32) {
    if (xor & (1<<j)  > 0)
        break

    j++
}
```

j

3) a = 0 b = 0

unset set

```
for ( i = 0        i < N        i++) {
        if (a[i]  &  (1 << j)  > 0)
                b = b ^ a[i]
        else
                a = a ^ a[i]
}
```

a → ✓

b → ✓

$i = 4$

$1 << i$

Check ith bit

1 or 0

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

> 0

$a \;\&\; (1<<i) > 0$

ith bit is 1

$a \;\&\; (1<<i) == 0$

ith bit is 0

2 5 101

$a = 56 = 111000$

$a-1 = 55 = 110111$

$a \,\&\, (a-1) = 110000$

$a \wedge a \,\&\, (a-1)$

last set bit

$a \wedge (a \,\&\, a-1)$

$$
\begin{array}{r}
111000 \\
\wedge\ 110000 \\
\hline
001000
\end{array}
$$