Decimal to Binary

$N = 25 \longrightarrow 11001$

2	25	1
2	12	0
2	6	0
2	3	1
2	1	1
	0	

$\log_2 N$

Binary to Decimal

$N = \underset{4}{1}\,\underset{3}{0}\,\underset{2}{1}\,\underset{1}{1}\,\underset{0}{0} = 2^4 + 2^2 + 2^1$

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

$= 22$

Bit wise operators (&, |, ^, ~, <<, >>)

a	b	a & b	a | b	a ^ b
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

a	~a
0	1
1	0

$a = 29 \quad 1\ 1\ 1\ 0\ 1$

$b = 18 \quad 1\ 0\ 0\ 1\ 0$

$a \& b \quad 1\ 0\ 0\ 0\ 0 \longrightarrow 16$

$a | b \quad 1\ 1\ 1\ 1\ 1 \longrightarrow 31$

$a \wedge b \quad 0\ 1\ 1\ 1\ 1 \longrightarrow 15$

$a \& b = b \& a$

$a \wedge b = b \wedge a$

$a | b = b | a$

$a \& b \& c = a \& c \& b = b \& c \& a$

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

$a | b | c = a | c | b = b | c | a$

$a = 10$

$10 : 1\ 0\ 1\ 0$

$|\ 1 : 0\ 0\ 0\ 1$

$1\ 0\ 1\ 1 \longrightarrow 11$

$a | 1$ — a is even → $a+1$; a is odd → a

$10 : \boxed{1\ 0\ 1}\ 0 \leftarrow$

$\&\ 1 : \boxed{0\ 0\ 0}\ 1$

$: 0\ 0\ 0\ 0$

$a \& 1$ — a is even → 0; a is odd → 1

$\boxed{a\,\hat{}\,0 = a}$

$$\begin{array}{rcl} 10 & : & 1\,0\,1\,0 \\ \hat{}\ \ 0 & : & 0\,0\,0\,0 \\ \hline 10 & : & 1\,0\,1\,0 \end{array}$$

$\boxed{a\,\hat{}\,a = 0}$

$$\begin{array}{rcl} 10 & : & 1\,0\,1\,0 \\ \hat{}\ 10 & : & 1\,0\,1\,0 \\ \hline & & 0\,0\,0\,0 \end{array}$$

<< & >>

$a = 5$	0	0	0	0	0	1	0	1		
$a<<1$	0	0	0	0	1	0	1	0	$\longrightarrow 10$	5×2^1
$a<<2$	0	0	0	1	0	1	0	0	$\longrightarrow 20$	5×2^2
$a<<3$	0	0	1	0	1	0	0	0	$\longrightarrow 40$	5×2^3
$a<<4$	0	1	0	1	0	0	0	0	$\longrightarrow 80$	5×2^4
$a<<5$	1	0	1	0	0	0	0	0	$\longrightarrow 160$	5×2^5
$a<<6$	0	1	0	0	0	0	0	0	$\longrightarrow 64$	5×2^6

Overflow

$\boxed{a << N = a \times 2^N}$

$\boxed{1 << N = 2^N} \Rightarrow O(1)$

$Pow(a, N) \Rightarrow a^N$

$\hookrightarrow \log N$

$Pow(2, N) \rightarrow 2^N$

$\hookrightarrow \log N$

$\gg$

a = 50	0	0	1	1	0	0	1	0	
a>>1	0	0	0	1	1	0	0	1	→ 25 : $50/2^1$
a>>2	0	0	0	0	1	1	0	0	→ 12 : $50/2^2$
a>>3	0	0	0	0	0	1	1	0	→ 6 : $50/2^3$
a>>4	0	0	0	0	0	0	1	1	→ 3 : $50/2^4$
a>>5	0	0	0	0	0	0	0	1	→ 1 : $50/2^5$
a>>6	0	0	0	0	0	0	0	0	→ 0 : $50/2^6$
a>>7	0	0	0	0	0	0	0	0	→ 0

$$\boxed{a >> N = a/2^N}$$

Q Given a no. N. Check if the i^{th} bit of N is set.

```
boolean checkBit(N, i) {
    if (((N>>i) & 1) == 1)
                ret true;
    else
        ret false;
}
```

N : 0 1 1 0 1 0 1 (bit positions 6 5 4 3 2 1 0)

i = 2 ⟶ T

i = 3 ⟶ F

N : 0 0 0 0 1 1 0 ~~1 0 1~~

& 0 0 0 0 0 0 1

Q Given a +ve no. Count the no. of set.

$N = 29$: 1 1 1 0 1 $\longrightarrow$ 4

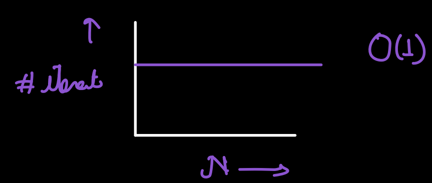
$N = 25$: 1 1 0 0 1 $\longrightarrow$ 3

$N = 15$: 0 1 1 1 1 $\longrightarrow$ 4

```
int countSetBits (N) {
    count = 0;
    for (i=0; i<32; i++) {
        if (checkBit(N, i)) {
            count ++;
        }
    }
    ret count;
}
```

TC : $\cancel{O(1)}$

$\underline{O(\log N)}$ }



Q Given an array where every no. appears twice except one no. Return the single no.

<u>XOR of all numbers</u>

TC : O(N)
SC : O(1)

$a \wedge a = 0$

$\cancel{a} \wedge \cancel{b} \wedge \cancel{a} \wedge \cancel{c} \wedge \cancel{b} \wedge \underline{d} \wedge \cancel{c}$

Google
CodeNation
InterviewBit/Scaler

Q Given an array where all no. appear twice except two numbers. $(N \geq 2)$
Return the two single no.

A : 3, 6, 4, 4, 6, 8 $\longrightarrow$ 3, 8

A : 4, 9, 9, 8 $\longrightarrow$ 4, 8

A : 1, 2 $\longrightarrow$ 1, 2

Approach 1 : Use HashMap
TC : O(N)
SC : O(N)

Approach 2 : Sort the array.

3, 3, 4, 4, 6, 8

TC : O(N logN)
SC : Depends on the sorting algo

MS $\rightarrow$ O(N)
QS $\rightarrow$ O(logN)

a, b, a, d, c, d, s_1, c, b, s_2

XOR of array $\longrightarrow$ $s_1 \wedge s_2$

XOR $\longrightarrow$ 7

$$\begin{array}{r} 000 \\ \wedge\ 111 \\ \hline 111 \end{array} \qquad \begin{array}{r} 010 \\ 101 \\ \hline 111 \end{array} \qquad \begin{array}{r} 100 \\ 011 \\ \hline 111 \end{array} \quad \cdots$$

Multiple possibilities

3, 6, 4, 4, 3, 8

XOR = 14

$$\begin{array}{r} 0\,1\,1\,0 \\ \wedge\ 1\,0\,0\,0 \\ \hline 1\,1\,1\,0 \\ \hline \end{array}$$

$0 \wedge 1$, $1 \wedge 0$ → 1

At all positions where the result of $S_1 \wedge S_2$ has a set bit

⟶ S_1 & S_2 have diff values at that bit.

$S_1 \wedge S_2 = 0$? ⟶ NO

$\because S_1 \,!= S_2$

$S_1 \wedge S_2 > 0$ (has at least one set bit)

A: 10 (1010), 8 (1000), 8 (1000), 9 (1001), 12 (1100), 9 (1001), 6 (0110), 11 (1011), 10 (1010), 6 (0110), 12 (1100), 17 (10001)

XOR of A → $17 \wedge 11$:

$$\begin{array}{r} 1\,0\,0\,0\,1 \\ \wedge\ 0\,1\,0\,1\,1 \\ \hline 1\,1\,0\,1\,0 \\ \hline \end{array}$$

	Set	Unset
Pos 1	10, 6, 11, 10, 6	8, 8, 9, 12, 9, 12, 17
Pos 3	10, 8, 8, 9, 12, 9, 11, 10, 12	6, 6, 17
Pos 4	17	10, 8, 8, 9, 9, 6, 11, 10, 6, 12, 12

Step I Take XOR of all array elements

(XOR stores $S_1 \wedge S_2$)

```
XOR=0;
for(i=0; i<N; i++){          O(N)
    XOR = XOR ^ A[i];
}
```

Step II Find position of any set bit in XOR

(Both single no will have diff val at this pos)

```
P=0;
while(XOR>0){
    if((XOR & 1)==1){
        break;               O(log Max)
    }
    XOR = XOR >>1;
    POS ++;
}
```

Step III Collect xor of all no. that have a set bit at position P in ans1 & unset bit at position P in ans2

```
ans1 = 0;   ans2 = 0;
for(i=0; i<N; i++){          O(N)
    if(CheckBit(A[i], P)){
        ans1 = ans1 ^ A[i];
    }
    else
        ans2 = ans2 ^ A[i];
}
```

ans1 $\longrightarrow S_1$

ans2 $\longrightarrow S_2$

TC: $O(N + \log Max + N)$

$\Rightarrow O(N + \log Max)$

SC: $O(1)$

ShareChat
LinkedIn

Q Given an array of size N containing all elements from $\boxed{1 \text{ to } N+2}$ except two no.
Find the missing no.

→ Without modifying the array
→ in constant space (SC: O(1))

A : 3 6 1 4 (indices 0 1 2 3) → 2, 5
(N=4) [1, 6]

A : 1 6 4 7 5 (indices 0 1 2 3 4) → 2, 3
N=5 [1, 7]

$2 \wedge 5 \rightarrow$ 010
101
110

set: 3, 1, 1, 3, 5
unset: 6, 4, 2, 4, 6

A : 3 6 1 4 (indices 0 1 2 3)
1 2 3 4 5 6

<u>Step I</u> Take XOR of all array elements. (XOR Array)

XOR Array $\rightarrow 3 \wedge 6 \wedge 1 \wedge 4$

<u>Step II</u> Iterate from 1 to N+2
take XOR in XOR Array.

XOR Array $\rightarrow 3 \wedge 6 \wedge 1 \wedge 4 \wedge 1 \wedge 2 \wedge 3 \wedge 4 \wedge 5 \wedge 6$

XOR Array $\rightarrow 2 \wedge 5$

Google
Amazon

Q Given an array. Every element appears thrice except one elements which appears once.
Find the single no.

* Without using any extra space

A : 5, 7, 5, 4, 7, 11, 11, 9, 11, 7, 5, 4, 4 $\longrightarrow$ 9

$a \wedge a = 0$

$a \wedge a \wedge a = a$

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

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

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

Hint

Can we find if the single no has a set/unset bit at i^{th} pos?

5, 7, 5, 4, 7, 11, 11, 9, 11, 7, 5, 4, 4

5	0	1	0	1
7	0	1	1	1
5	0	1	0	1
4	0	1	0	0
7	0	1	1	1
11	1	0	1	1
11	1	0	1	1
9	1	0	0	1
11	1	0	1	1
7	0	1	1	1
5	0	1	0	1
4	0	1	0	0
4	0	1	0	0
	4	9	6	10

1001 $\Rightarrow$ 9

Approach

Iterate over all bit positions
for every bit position
count no of set bits.

Count % 3 == 0 ⟶ single no has unset bit
Count % 3 != 0 ⟶ single no has set bit

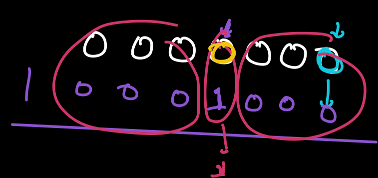
```
ans = 0;
for (i=0; i< 32; i++) {        // 32 → int , 64 → long
    // count no of set bits at pos i
    count = 0;
    for (j=0; j<N; j++){        // O(N)
        if (checkBit( A[j], i)){
            count++;
        }
    }

    if ( count % 3 != 0) {
        // i-th bit of single no is set
        // hence set the i-th bit in ans.
        ans = ans | (1<<i);
    }
}
ret ans;
```

7 ___

15 ____

A[j] , i

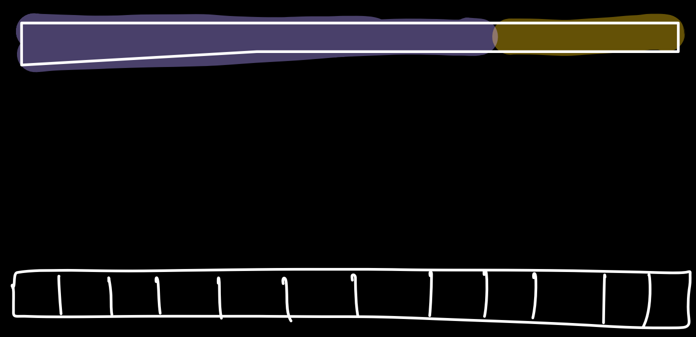
ans = 0 ;



int ⟶ 2^{32} ⟶ 32
long ⟶ 2^{64} ⟶ 64

TC : $O(N \log Max)$ ⟶ Max of datatype

SC : $O(1)$



$\boxed{f > N/2}$

$f > N/3$