Single Number 1

Every no. appears 2 times & one no. appears once
Find that no.

{ 4, 5, 5, 4, 1, 6, 6 }
{ 7, 5, 5, 1, 7, 6, 1, 6, 4 }

Brute Force:

Iterate through entire array & get the count for every no.

TC: $O(N^2)$ SC: $O(1)$

Hashmap.

TC: $O(N)$ SC: $O(N)$

~~120~~ ^ ~~5~~ ^ ~~6~~ ^ ~~6~~ ^ ~~120~~ ^ ~~5~~ = 0

A ^ A = 0

Optimised Approach 1:

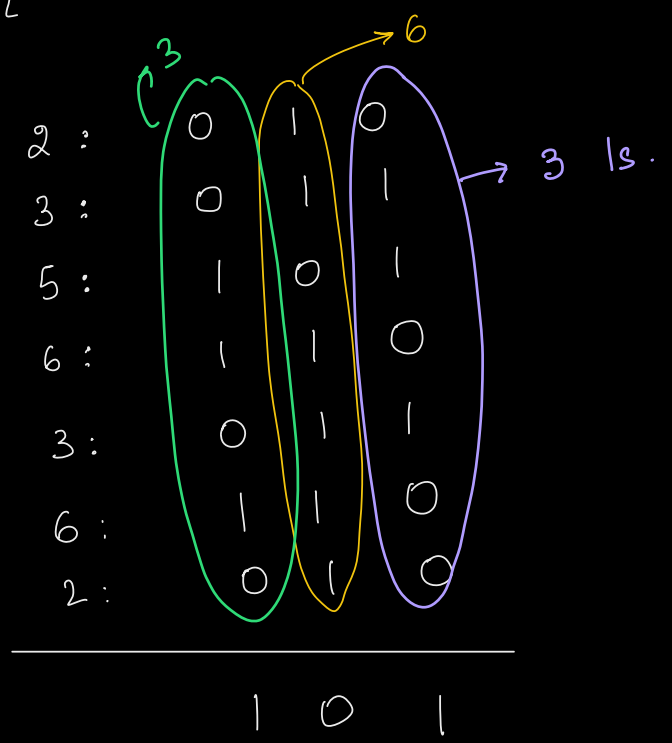
Take XOR of all elements.

4 ^ 5 ^ 5 ^ 1 ^ 4

```
int x = 0 ; // x^0 = x
for(i=0; i<N; i++){
    x = x ^ A[i]
}
print(x)    // single no.
```

Optimised Approach 2:

{ 2 3 5 6 3 6 2 }




```
ans = 0 ;
for ( i=0 ; i<32 ; i++) {
    int cnt = 0
    for ( j=0 ; j<N ; j++) {
        if ( checkBit ( A[j], i )) {
            cnt ++
        }
    }
    if (cnt %2 != 0) {
        ans = ans | (1<<i)
    }
}
```

(A[j]: No. , i: bit)

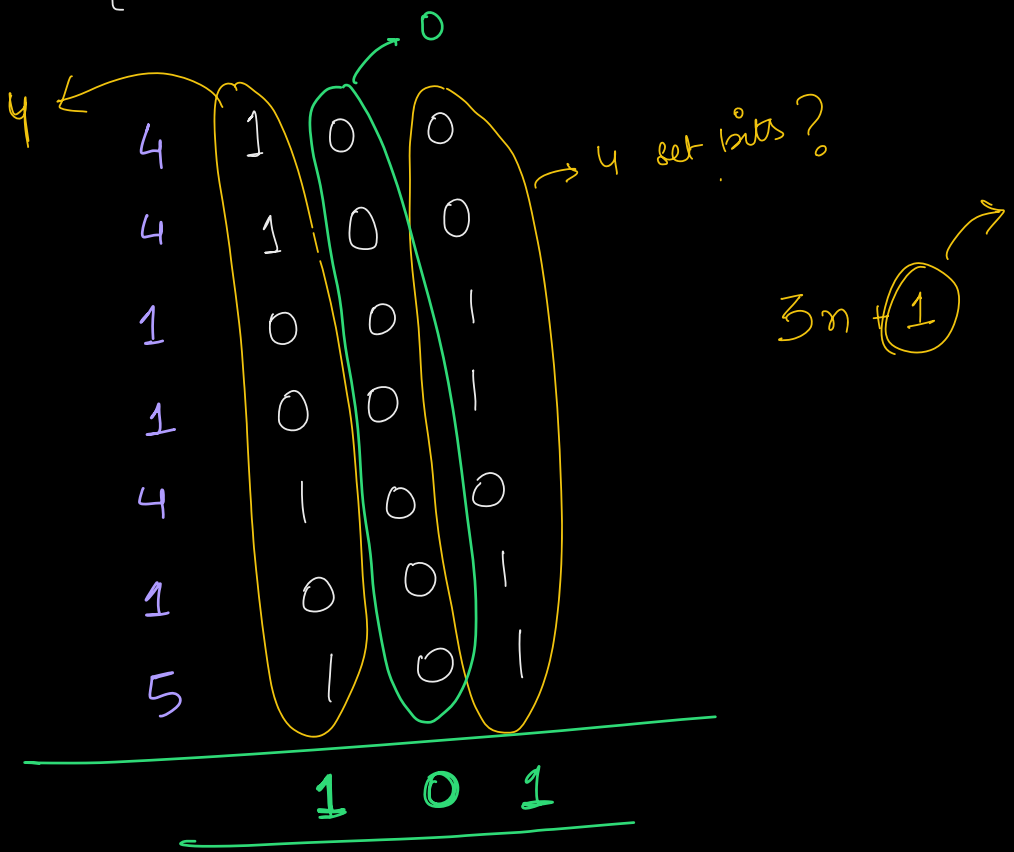
TC: O(N)

SC: O(1)

Single Number 2.

Every no. appears 3 times & one no. appears once
Find that no.

Q. { 4 4 1 1 4 1 5 }



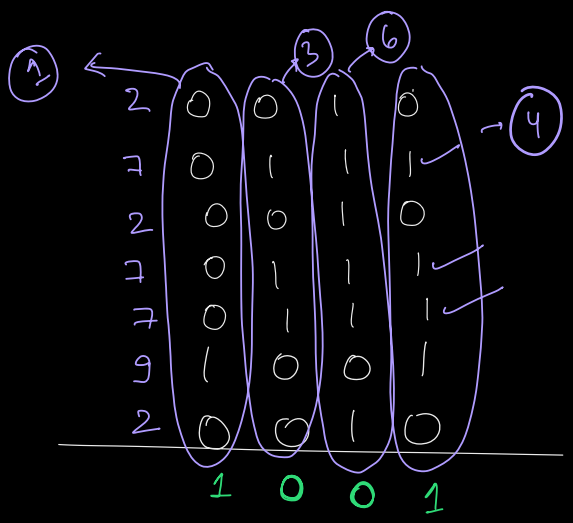
```
ans = 0 ;
for ( i=0 ; i<32 ; i++) {
    int cnt = 0
    for ( j=0 ; j<N ; j++) {
        if ( checkBit ( A[j], i )) {
            cnt ++
        }
    }
    if (cnt % 3 != 0) {
        ans = ans | (1<<i)
    }
}
```

(NO. = A[j], bit = i)

TC: O(N)

SC: O(1)

{ 2 7 2 7 7 9 2 }



Q3. Single Number 3

Given an array, all elements appear twice except two numbers. Find those 2 nos. They appear only once.

{5, 4, 7, 6, 7, 4, 5, 2}

6 ^ 2 = 4

From XOR of 2 nos., how can we identify the nos.?

{1, 1, 7, 6, 7, 5, 5, 2}

6 ^ 2 = 4

3	2	1	0
0	1	1	0
0	0	1	0
0	1	0	0

why is this set? Because 2nd bit is different in 6 & 2

{ 4 9 5 4 9 3 }

5 ^ 3 = 6

0 1 0 1
0 0 1 1
---
0 1 1 0

Based on any of set bits, we will be able to distinguish b/w the 2 nos. giving this XOR

0100 1001 0101 0100 1001 0011

{ 4 9 5 4 9 3 }

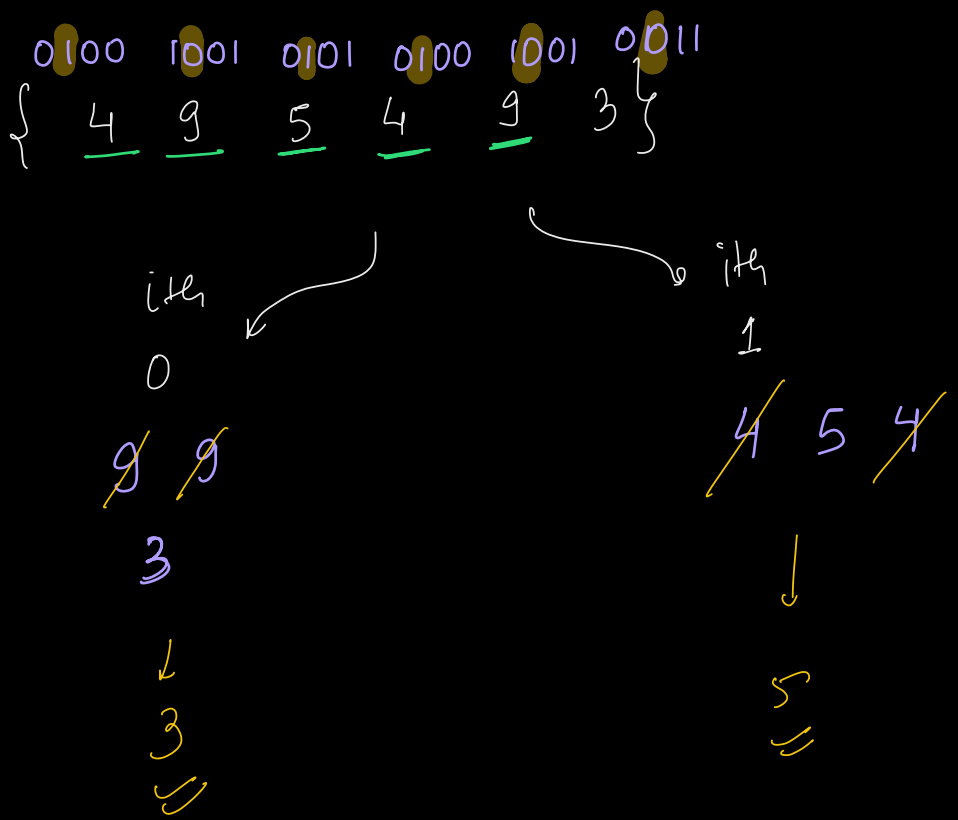
ith 0

Based on ith bit can we distribute ele in 2 sets?

ith 1

4 9

5 4 9


0100 1001 0101 0100 1001 0011
{ 4 9 5 4 9 3 }
ith
0
9 9
3
3
ith
1
4 5 4
!
5

Pseudocode

```
int x = 0
for( i=0; i<N; i++){
|     x = x ^ A[i]
}
```

N

TC: O(N)

SC: O(1)

// x contains XOR of 2 unique nos., all nos. appearing in pairs got cancelled out

```
for( pos=0 ; pos<32 ; pos++){
|     if ( checkBit( x , pos )) {
|     |     break
|     }
}
```

32

// pos will contain the bit where 1 is present

```
x1 = 0   // where pos bit is set
x2 = 0   // unset

for( i=0; i<N; i++){
|     if ( checkBit ( A[i], pos )) {
|     |     x1 = x1 ^ A[i]
|     }
|     else{  x2 = x2 ^ A[i] }
}
print(x1 , x2) ;
```

N

Break : 8:50

Maximum AND pair

Given array, find pair whose AND is maximum

$i != j$

$\{ \underset{0}{5}, \underset{1}{4}, \underset{2}{6}, \underset{3}{8}, \underset{4}{5} \}$ → Ans $(0, 4)$ or $(4, 0)$

$5 \& 4$ $4 \& 6$ $6 \& 8$
$5 \& 6$ $4 \& 8$ $6 \& 5$
$5 \& 8$ $4 \& 5$ $8 \& 5$
$5 \& 5 = (5)$

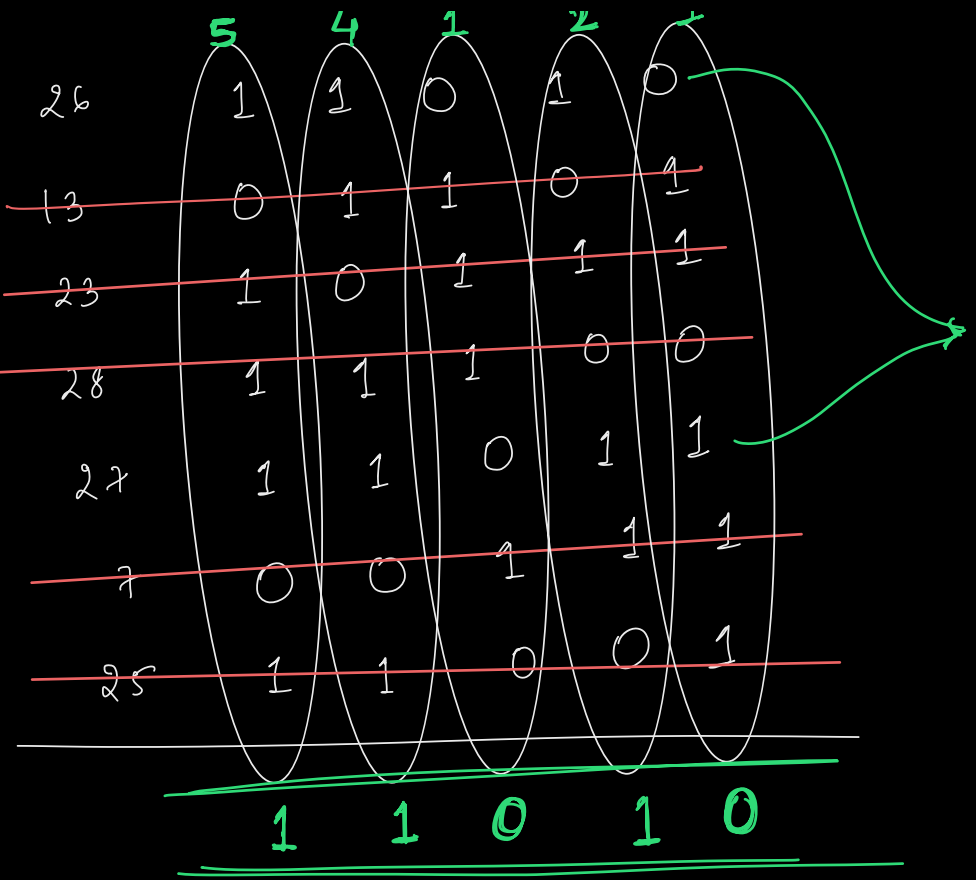
$\{ \underset{10101}{21}, \underset{10010}{18}, \underset{11000}{24}, \underset{10001}{17}, \underset{10000}{16}, \underset{00111}{7}, \underset{00011}{3} \}$

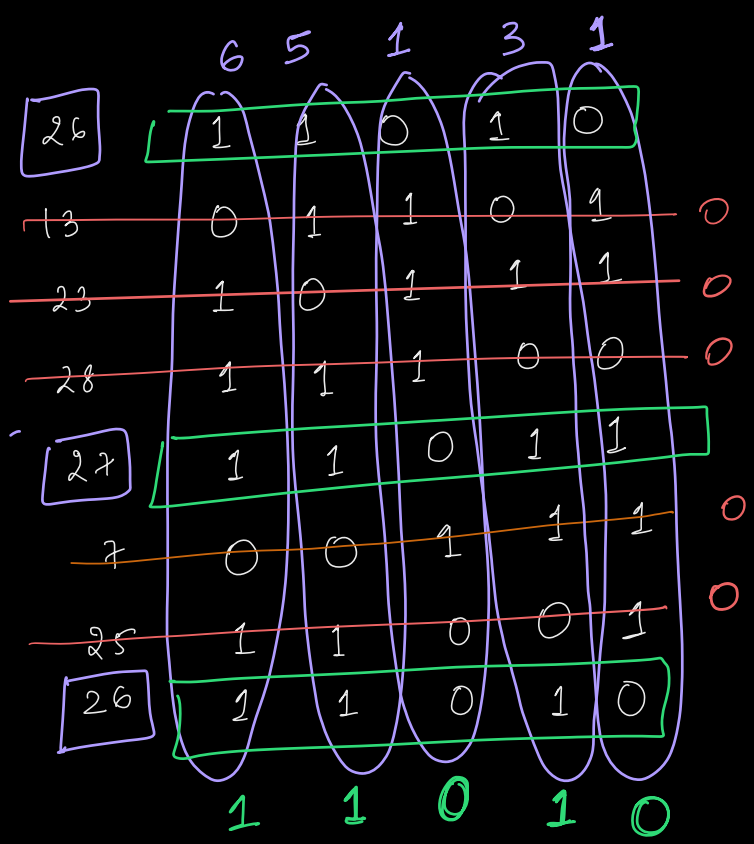
$\{ \underset{0101}{5}, \underset{0100}{4}, \underset{0011}{3}, \underset{0010}{2}, \underset{0001}{1} \}$

We want 1 to be present as left as possible in the answer

$\{26, 13, 23, 28, 27, 7, 25\}$

26	1	1	0	1	0
~~13~~	0	1	1	0	1
~~23~~	1	0	1	1	1
~~28~~	1	1	1	0	0
27	1	1	0	1	1
~~7~~	0	0	1	1	1
~~25~~	1	1	0	0	1
	1	1	0	1	0





$$NC_2 = \frac{N!}{2!(N-2)!}$$

$$= \frac{(N-1) \times N}{2}$$

$$\frac{(3-1) \times 3}{2}$$

$$= 3$$

26 27
26 26
27 26

Code

```
ans = 0
for ( i = 31 ; i >= 0 ; i--) {
    cnt = 0
    for ( j = 0 ; j < N ; j++){
        if ( checkBit (A[j], i)) {
            cnt++
        }
    }
    if ( cnt >= 2){
        ans = ans | (1 << i)
        for( K = 0; K < N; K++){
            if( checkBit (A[K], i) == 0){
                A[K] = 0
            }
        }
    }
}
print (ans);
```

TC: O(32 * N)

O(N)

SC: O(1)