6

110
& 1
0

ans = 0

while(n>0){
int rem = n & 1
ans = rem + ans
n >>= 1
}

a+b

[1 2 3]

1   2   3
1 2   2 3
1 2 3

=> (1) ^ (1^2) ^ (1^2^3) ^ (2) ^ (2^3) ^ (3)

=> 1^1^1^2^2^2^2^3^3^3 = 1^3

(a^b) = (b^a)

a^0 = a

a^a = 0

5 8 9 8 5
1, 2, 3, 4, 5 => 1^3^5

001
011
101
111 => 7

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

2
2^3
2^3^4
2^3^4^5

3
3^4
3^4^5

4
4^5

5

5 [7 9 6 4 3 2]
0 1 2 3 4 5 6

freq = (i+1) * (n-i)

1 → 6
0 → 6

s = 101
& 1
1

6 = 110
7 = 111

32 logn

$(\log n)^2$

```java
public static void displayBinary(int n){
    String s= "";
    for(int i = 0;i<32;i++){
        int res = (n & (1<<i)) ;
        if(res != 0) s ="1"+ s;
        else s= "0"+s;
    }
    System.out.println(s);
}
```

00001100
& 00100
00000100

i=2

s = 100

1024 => 10

$2^{10} = 1024$

$\log_2 1024 = 10$

$(\log n)^2$

32

1000000000 → 31

1024

1	0	0	0	0	0	0	0	0	0	0
1024	512	256	128	64	32	16	8	4	2	1

4 2 4 3 2 5 3 2 3 4

// I want to keep track of the freq of every ith bit

int[] freq = new int[32]

for(int x : arr){
for(int i=0;i<32;i++){
if(ith bit is set(x, i)) freq[i]++
}
}

ans=0

for(int i = 0; i<32; i++){
if(freq[i] % 3 != 0){
ans += (1<<i);
}
}

$10^6 \cdot 32$

O(32n)

TC: O(n)
SC: O(1)

-5 → -4

2^0 => 1 (1<<0) => 1

2^1 => 2 (1<<1) => 2

2^2 => 4 (1<<2) → 100 => 4

2^3 => 8 (1<<3) → 1000 => 8

2^{bit}

101101
001000
001000

10110101
00010000
00010000