

## ① One Odd Occurance.

↳  $[1, 2, 3, 2, 3] \Rightarrow$  only one occurs once.

↳  $0 \wedge x = x$      $x \wedge x = 0$

so start with  $res = 0$

$$0 \wedge 1 \Rightarrow 1$$

$$1 \wedge 2 \Rightarrow 1 \wedge 2$$

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

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

$$1 \wedge 3 \wedge 3 \Rightarrow 1$$

## ② 2 odd Occurances.

↳ all even occurrences except 2 numbers.

$$[5, 6, 10, 6, 6, 10, 3, 3]$$

①  $xor \Rightarrow 5 \wedge 6 \Rightarrow 0101 \wedge 0110 \Rightarrow 3$

②

→ Find XOR of all numbers

→ How to find the numbers from result ??

As we know the set in the XOR result means it's different in both numbers.

So consider any index which is a set and also consider which is unset.

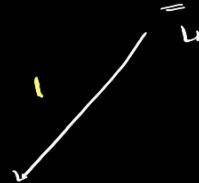
and XOR them separately.

For extracting the right most set bit which is last set bit

$$k = res \& \sim(res - 1)$$

↑

$$3 \Rightarrow 11$$



Any bit which is set (Right most)

(For simplicity we can take the last one which is set bit of the num.)

$$5 \wedge 3 \wedge 3$$

$$= 5 \checkmark$$

$$6 \wedge 10 \wedge 6 \wedge 10 \wedge 6 = 6 \checkmark$$

③ Given a number, we need to find XOR of numbers  
 from 1 to N.

if we use a loop  
 $\rightarrow O(N)$

$O(1)$

$\Downarrow$

if  $n \% 4 == 0$  :  
 $\text{print}(n)$

$n \% 4 == 1$  :  
 $\text{print}(1)$

$n \% 4 == 2$  :  
 $\text{print}(n+1)$

$n \% 4 == 3$  :  
 $\text{print}(0)$

$n = 1$	1	
$n = 2$	3	
$n = 3$	0	
$n = 4$	4	$\Rightarrow$
$n = 5$	1	
$n = 6$	7	
$n = 7$	0	
$n = 8$	8	

④ Given L, R we need to compute XOR from L to R.

Consider  $L = 3$   
 $R = 5 \Rightarrow 3 \wedge 4 \wedge 5$   $O(1)$

$\sim$  XOR till 5 =  $1 \wedge 2 \wedge 3 \wedge 4 \wedge 5$

XOR till 2 =  $1 \wedge 2$

$\Rightarrow$  XOR till 5  $\wedge$  XOR till 2.

$\Rightarrow$  clear Last set Bit

1 1 0 1 1 0

$\Rightarrow$  1 1 0 1 0 0

12  $\Rightarrow$  11 00  
 11  $\Rightarrow$  10 11  
 10 00 ✓

312  $\Rightarrow$  11 10  
 10 10  $\Rightarrow$  42

413  $\Rightarrow$  100 11  
 100 000  $\Rightarrow$  40

→ set last unset bit

$$4 \Rightarrow \begin{array}{r} 100 \\ (0) \underline{101} \\ 101 \end{array} \rightarrow 5 \quad \bigg| \quad \begin{array}{r} 5 \quad 101 \\ 0 \quad \underline{110} \\ (6) \end{array}$$

for setting last bit  $\Rightarrow n \mid n+1$

$$9 \Rightarrow \begin{array}{r} 1001 \\ 1010 \\ \underline{\phantom{0000}} \\ 1011 \end{array} \rightarrow 11 \checkmark$$

→ Power set using bitwise ..

str = 'abc'

n = 3

↳ no. of subsets =  $2^n = 2^3 = 8$ .

These subsets can be mapped to binary representations of numbers from 0 - 7.

	c	b	a			
0	0	0	0	c	5	1 0 1
1	0	0	1	a	6	1 1 0
2	0	1	0	b	7	1 1 1
3	0	1	1	ab		
4	1	0	0	c		

→ compute power of 2 (For running loop)  
→ run a loop from  $0 \rightarrow 2^n - 1$

for i in range( $2^n$ ):

subst = []

for j in range(n):

if  $((i \& (1 << j)) \neq 0)$ :

add to subst

add to power set.

