Bit Manipulation Problems

Todays Content

- Bitwise operators & properties
- Check if bit is set or unset
- Single element - I
- Single element - II
- Single element - III
- Max AND Pair

A	B	A & B	A | B	A ^ B	~A	~B
0	1	0	1	1	1	0
0	0	0	0	0	1	1
1	1	1	1	0	0	0
1	0	0	1	1	0	1

1 set
0 unset

A & B – Both A and B must be 1, to get 1
A | B – One of A or B must be 1 to get 1
A ^ B – A != B , then ans is 1

same some puppy shame.

Bitwise Operations

A = 20
B = 45

		7	6	5	4	3	2	1	0	
A	=	0	0	0	1	0	1	0	0	
B	=	0	0	1	0	1	1	0	1	
Print A & B	=	0	0	0	0	0	1	0	0	= 4
Print A | B	=	0	0	1	1	1	1	0	1	= 61
Print A ^ B	=	0	0	1	1	1	0	0	1	= 57

Properties

Commutative

a & b = b & a
a | b = b | a
a ^ b = b ^ a

Associative

a & b & c = b & a & c = c & a & b
a | b | c = b | a | c = c | a | b
a ^ b ^ c = b ^ a ^ c = c ^ a ^ b

* a & 1 →(1) a is odd
a & 1 →(0) a is even

If the 0th bit is 1 → 1 , else 0

```
4   100        5    101
  & 001           & 001
    000             001
```

$a \,\&\, o = 0$	$a \mid o = a$	$a \wedge o = a$
$a \,\&\, a = a$	$a \mid a = a$	$a \wedge a = 0$

$$\begin{array}{rccccc} & 1 & 0 & 1 & 1 & 0 \\ \wedge & 0 & 0 & 0 & 0 & 0 \\ \hline & 1 & 0 & 1 & 1 & 0 \\ \hline \end{array}$$

$$\begin{array}{rcc} a & 0 & 1 \\ & 0 & 0 \\ \hline a & 0 & 1 \end{array}$$

Left & Right Shift operator.

Right Shift >>

a = 50	0	0	1	1	0	0	1	0	50
a >> 1	0	0	0	1	1	0	0	1	25
a >> 2	0	0	0	0	1	1	0	0	12

$$a >> i = \frac{a}{2^i}$$

Left shift <<

$a = 5$	0	0	0	0	0	1	0	1	5
$a << 1$	0	0	0	0	1	0	1	0	10
$a << 2$	0	0	0	1	0	1	0	0	20

$$a << i = a \times 2^{i}$$

$$1 << i = 2^{i}$$

Q1> Given a number N and i . Check if the i^{th} bit is set or not

$N = 53$
$i = 0$

$$\begin{array}{ccccccc} 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{array}$$

if N & 1 == 1
        1
else
        0

$N = 53$
$i = 1$

$$\begin{array}{ccccccc} 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ & 0 & 1 & 1 & 0 & 1 & 0 \end{array}$$

if (N>> 1) & 1
        1
else
        0

$N = 53$
$i = 2$

$$\begin{array}{ccccccc} 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{array}$$

$\boxed{N>>2}$

(N>> i) & 1 == 1

i

```
bool   checkBit ( N , i )  {
   |
   |   return  (N>>i) & 1 == 1
   |
}
```

Single Element I

Every element repeats twice except 1. Find the single element ?

Eg : A[7] = {3 2 3 7 2 8 7} → 8

A[9] = {3 6 2 3 5 4 5 6 4} → 2

- Brute force :

  For each element
      count its freq and if its == 1 return

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

- Idea 2 : Freq map → Check for which ele freq==1

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

- Idea 3 : XOR every value of array.

```
ans = 0
for i 0 → n-1 {
|     ans = ans ^ A[i]
}
    return ans
```

Single Element II

Given A[N], every element repeats thrice except 1
Find the unique element?

Constraints : $A[i] \rightarrow (1, 10^9)$
$N \rightarrow (1, 10^5)$

$A[7] = \{6\ 5\ 6\ 4\ 5\ 6\ 5\} \rightarrow 4$

$A[13] = \{5\ 7\ 5\ 4\ 7\ 11\ 11\ 9\ 11\ 7\ 5\ 4\ 4\} \rightarrow 9$

- Brute force :

    For each element
    count its freq and if its == 1 return

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

- Idea 2 : Freq map → Check for which ele freq==1

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

- Idea 3 : Sort and check adj elements.

TC : O(nlogn)
SC : O(1)

2 2 2 3 4 4 4

A[13] = { 5 7 5 7 11 11 9 11 7 5 }

		3	2	1	0
5	:	0	1	0	1
5	:	0	1	0	1
5	:	0	1	0	1
7	:	0	1	1	1
7	:	0	1	1	1
7	:	0	1	1	1
11	:	1	0	1	1
11	:	1	0	1	1
11	:	1	0	1	1
9	:	1	0	0	1
Total		3^{+1}	6^{+0}	6^{+0}	9^{+1}
		% 3	% 3	% 3	% 3
		1	0	0	1
		3	2	1	0

$ans = 0$

$= +1$

$= +2^3$

$ans = ans + 2^i$

$ans + (1 << i)$

Observation

c = count of i^{th} set bits.

$c \% 3 == 1$

set i^{th} bit in the ans

```
int  singleII ( A[] ) {

    ans = 0
    for ( bit → 0 to 31 ) {
        c = 0
        for ( i → 0 to n-1 ) {

            if ( checkBit ( A[i], bit ) ) {
                    c += 1
            }
        }

        if ( c%3 == 1 ) {
            ans += 1 << bit
        }
    }
    return ans
}
```

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

Extensions

→ Every element repeats thrice except 1 element which repeats 2 times

$(c \% 3 == 2)$ set the ith bit in ans.

→ Every element repeats 4 times

→ Except 1 element, repeats 1 time

Xor all elements.

→ Except 1 element, repeats 2 time

$(c \% 4 == 2)$

→ Except 1 element, repeats 3 time

Xor all elements.

Break → 10:00 pm

- ## Single Element III

Given arr[N], every element repeats twice except for 2 elements. Find the two unique elements.

Eg: $A[6] = \{3\ 6\ 4\ 4\ 3\ 8\} = \rightarrow (6,8)$

$A[4] = \{4\ 9\ 9\ 8\} = (4,8)$

- Brute force :

  For each element
      count its freq and if its == 1 return

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

- Idea 2 : Freq map ⟶ Check for which ele freq==1

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

- Idea 3 : Sort and check adj elements.

TC : $O(n \log n)$
SC : $O(1)$

2 2 3 5 4 4

- Idea 4 : Xor all the elements. ✗

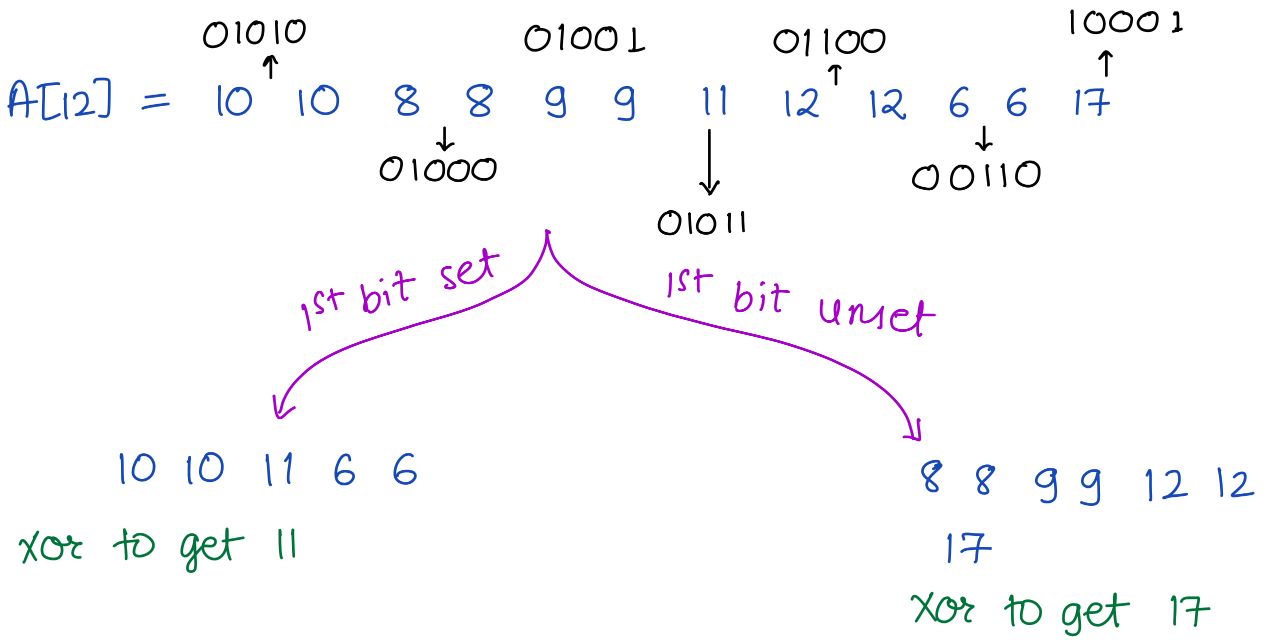
~~2~~ ~~2~~ 3 5 ~~4~~ ~~4~~

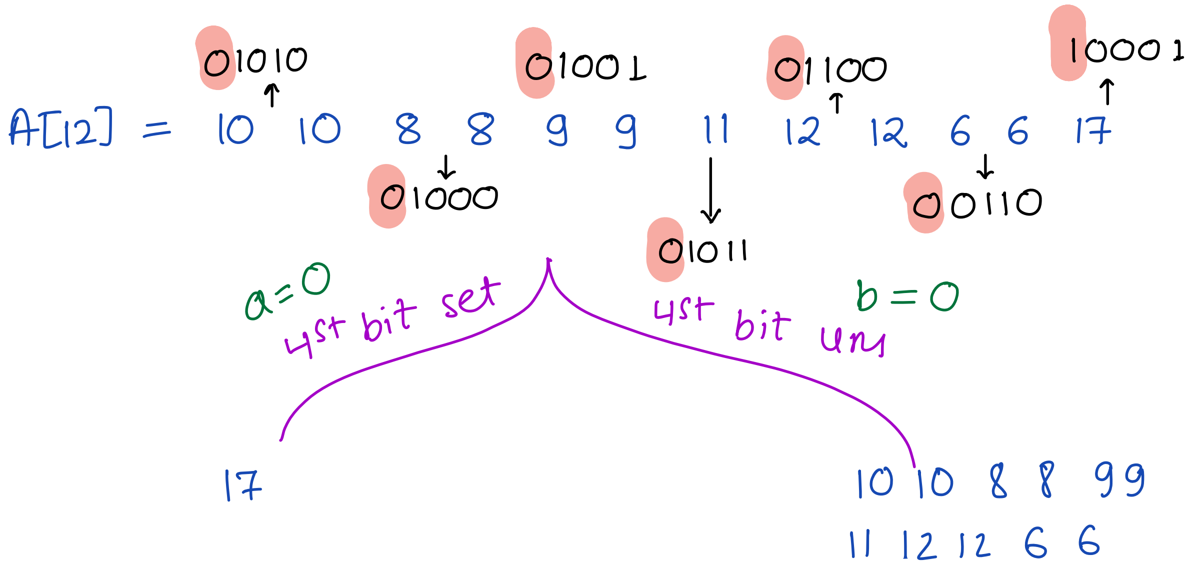
$3 \wedge 5$

$$\begin{array}{r} 101 \\ 011 \\ \hline 110 \\ \boxed{6} \end{array}$$

```
11 :   0 1 0 1 1
17 :   1 0 0 0 1
---------------
       1 1 0 1 0
       4 3 2 1 0
```

Observation : if xor of 2 bits is 1.
They are different at that bit position.





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

Pseudocode :

```
singleIII ( A[] ) {
        a = 0 , b = 0
    // Step 1  Xor all the elements.
        v = 0
        for ( i -> 0 to n-1) {
        |     v ^= A[i]
        }
    // Step 2  Find any set bit position
        pos = 0
        for ( bit 0 -> 31 ) {
        |   if (checkbit( v, bit)) {
        |   |     pos = bit ; break
        |   }
        }
    // Step 3  if pos^th bit is set xor with a else b
        for ( i -> 0 to n-1) {
            if (checkBit (A[i], pos) {
            |     a ^= A[i]
            }
            else {
                  b ^= A[i]
            }
} return a, b
```

Q> Given A[N], Choose two indices (i, j) such that $(i \neq j)$ and A[i] & A[j] is maximum. {Google}
Return the max value.

A[3] = { 27 18 20 }

27	=	1	1	0	1	1
18	=	1	0	0	1	0
20	=	1	0	1	0	0

27 & 18 = 18
27 & 20 = 16
18 & 20 = 16

Bruteforce :

for each pair do & and return max

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

sort and check adj pairs

18 20 27

16 16

✗

27 17 0

A[7] = 26 13 23 28 27 7 25

		4	3	2	1	0
26	=	1	1	0	1	0
~~13~~ 0	~~=~~	~~0~~0	~~1~~0	~~1~~0	~~0~~0	~~1~~0
~~23~~0	~~=~~	~~1~~	~~0~~	~~1~~	~~1~~	~~1~~
~~28~~	~~=~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~
27	=	1	1	0	1	1
~~7~~ 0	~~=~~	~~0~~	~~0~~	~~1~~0	~~1~~0	~~1~~0
~~25~~	~~=~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~
		4	3	2	1	0
		5	4	1	2	1

continue

⇒ $2^4 + 2^3 + 2^1 = 16 + 8 + 1$

$= 25$

	4	3	2	1	0
~~7~~	~~0~~	~~0~~	~~1~~	~~1~~	~~1~~
11	0	1	0	1	1
25	1	1	0	0	1
		3	2	1	0
	1	2	0	1	2

ignore
no pair

1

$8 + 1 = 9$

```
int  maxAnd ( A[]) {

    ans = 0

    for ( bit -> 31 to 0 ) {
        c = 0
        for ( i -> 0 to n-1 ) {
            if ( CheckBit ( A[i] , bit) {
                c++;
            }
        }

        if ( c > 1) {
            ans +=  1 << bit
            // set TO 0
            for ( i -> 0 to n-1 ) {
                if (! CheckBit (A[i] , bit) {
                    A[i] = 0
                }
        }
    }
    return ans
}
```

TC : O(N)

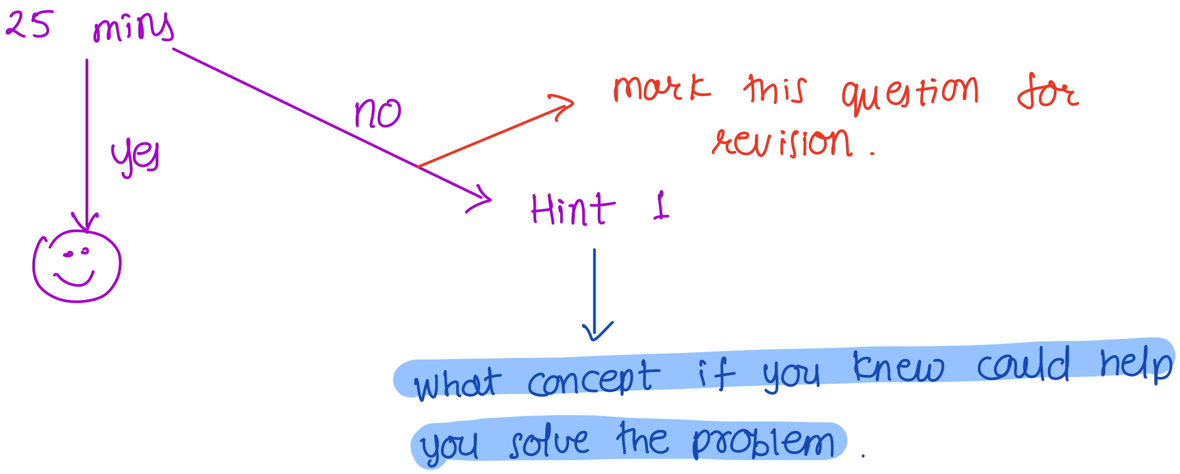
SC : O(1)

Doubt.

A & 1

```
a = 10

if (a & 1 == 1)
```



mark this question for revision.

what concept if you knew could help you solve the problem.