Q → Given an integer array where every element appears Twice except for one element that appears once. Find that unique element.

A = [2 3 5 6 3 8 2] → 5

```
ans = 0
for i → 0 to (N-1)
    ans ^= A[i]
return ans
```

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

Q → Given an integer array where every element appears Twice except for two element that appears once. Find the two unique elements.

A = [2 3 2 5 3 6 7 6] Ans = {5, 7}
A = [1 2 3 1] Ans = {2, 3}

Bruteforce → ∀ A[i] check if it is unique.

- Travel: TC = O(N²), SC = O(1)
- Frequency: TC = O(N) ✓, SC = O(N)

XOR of all elements

A = [2 3 2 5 3 6 7 6]

2^3^2^5^3^6^7^6 = 5^7 = 2

5 → 1 0 1
7 → ^ 1 1 1
0 1 0 → 2

4^6 = 2
9^11 = 2
8^10 = 2

(x != y) x^y > 0

A = [9 11 9 8 10 11] → 9^11^9^8^10^11 = 8^10 = 2 > 0

Ans → x, y

XOR of all numbers → $x \wedge y = a > 0$ (at least one set bit)

set bit in a is either set bit in x or y & not both.

$a = 12 \rightarrow (1100)_2$

$10 \rightarrow (1010)_2$

$4 \rightarrow (0100)_2$

$13 \wedge 8 = 5$

$13 \rightarrow 1101$

$8 \rightarrow 1000$

0101

use any set bit in $x \wedge y$ to divide all elements in two groups.

A = [2 4 2 5 4 6 7 6]

XOR of all → $2 \wedge 4 \wedge 2 \wedge 5 \wedge 4 \wedge 6 \wedge 7 \wedge 6 = 5 \wedge 7 = 2$ $(00\overset{3}{\underset{}{}}\overset{2}{\underset{}{}}\overset{1}{\underset{}{}}\overset{0}{\underset{}{}}10)_2$

b = 1st bit is set → {2, 2, 6, 7, 6}

unset → {4, 5, 4}

$4 \wedge 5 \wedge 4 = 5$ ✓

$2 \wedge 2 \wedge 6 \wedge 7 \wedge 6 = 7$ ✓

$2 \rightarrow 0010$ (1)

$4 \rightarrow 0100$

$5 \rightarrow 0101$

$6 \rightarrow 0110$

$7 \rightarrow 0111$

Steps

1) XOR of all elements

```
xor = 0
for i → 0 to (N-1)
    xor ^= A[i]
```

2) Find any set bit in xor

```
b = -1        // A[i] <= 10^9
for i → 0 to 30
    if ((xor >> i) & 1 == 1)
        b = i
        break
```

3) Take XOR of two groups separately

```
x = 0    y = 0
for i → 0 to (N-1)
    if ((A[i] >> b) & 1 == 1)
        x ^= A[i]
    else
        y ^= A[i]
print (x, y) ✓
```

TC = O(N + 31 + N) = O(N)

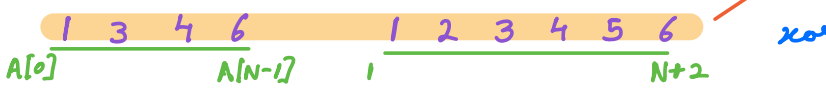
SC = O(1)

$2^{30} > 10^9$

Q → Given an integer array of N length where elements from 1 to N+2 are present exactly once except for two elements. Find the two missing elements.

A = [3 6 1 5 4] N = 5 [1 7] Ans = {2, 7}

A = [1 3 4 6] N = 4 [1 6] Ans = {2, 5}



xor = 2^5 = 7 $(0\ 1\ 1\ 1)_2$

Steps

1) XOR of all elements

```
xor = 0
for i → 0 to (N-1)
    xor ^= A[i]
for n → 1 to (N+2)
    xor ^= n
```

2) Find any set bit in xor

```
b = -1        // A[i] <= 10^9
for i → 0 to 30
    if ((xor >> i) & 1 == 1)
        b = i
        break
```

3) Take XOR of two groups separately

```
x = 0    y = 0
for i → 0 to (N-1)
    if ((A[i] >> b) & 1 == 1)
        x ^= A[i]
    else
        y ^= A[i]
for n → 1 to (N+2)
    if ((n >> b) & 1 == 1)
        x ^= n
    else
        y ^= n
print (x, y) ✓
```

A = [3 6 1 5 4] N = 5 N+2 = 7

1) xor = $3^6^1^5^4^1^2^3^4^5^6^7 = 2^7 = 5 \rightarrow (0101)_2$ (b = 0)

2) b = 0 ✓

3) x = $3^1^5^1^3^5^7 = 7$ ✓ (1 → 0001, 5 → 0101)

y = $6^4^2^4^6 = 2$ ✓ (2 → 0010, 6 → 0110)

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

3 → 0 0 1 1 7 → 0 1 1 1

4 → 0 1 0 0

Q → Given an integer array, find max value of (A[i] & A[j]) s.t $i \neq j$

A = [16, 9, 11, 10] (indices 0 1 2 3)

(A[i] & A[j] = A[j] & A[i])

Bruteforce → ∀ i, j s.t $i \neq j$ find A[i] & A[j] and store max value.

$i < j$

16 & 9 = 0 9 & 11 = 9 16 → 1 0 0 0 0

16 & 11 = 0 9 & 10 = 8 9 → 0 1 0 0 1

16 & 10 = 0 11 & 10 = 10 (Ans) 11 → 0 1 0 1 1

TC = O(N^2) SC = O(1) 10 → 0 1 0 1 0

A = [10 4 6 12 13 16] (indices 0 1 2 3 4 5) → Ans = 12 & 13 = 12 ✓

2^4

$(1\,0\,0\,0\,0)_2 > (0\,1\,1\,1\,1)_2$ (bits 4 3 2 1 0)

16 > 15

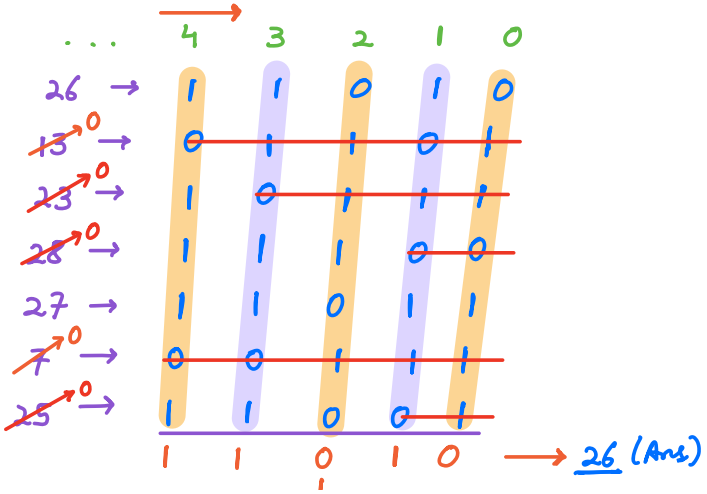
$2^3+2^2+2^1+2^0 = 2^4-1 = 15$

bits on left side have more preference.

$2^0+2^1+2^2 \ldots 2^K = 2^{K+1}-1$

AND → x & y → i^{th} bit is 1 if it is 1 in x as well as y.

A = [26 13 23 28 27 7 25]



↓
only 1 remaining no. has this bit set.
to make it set, atleast 2 numbers should have this bit 1.

// A[i] <= 10^9

```
ans = 0
for b → 30 to 0
    // count # A[i] where b^th bit = 1
    cnt = 0
    for i → 0 to (N-1)
        cnt += (A[i] >> b) & 1

    if (cnt >= 2) {
        ans |= (1 << b)   // set b^th bit in ans
        for i → 0 to (N-1)
            if ((A[i] >> b) & 1 == 0)
                A[i] = 0
    }
return ans
```

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

12:55 PM

Q → Given an integer array of N +ve elements.
Calculate sum of XOR of all pairs A[i] ^ A[j] s.t $\boxed{i < j}$

A = [3 5 6] (0 1 2)

3 ^ 5 = 6
3 ^ 6 = 5
5 ^ 6 = 3
14 (Ans)

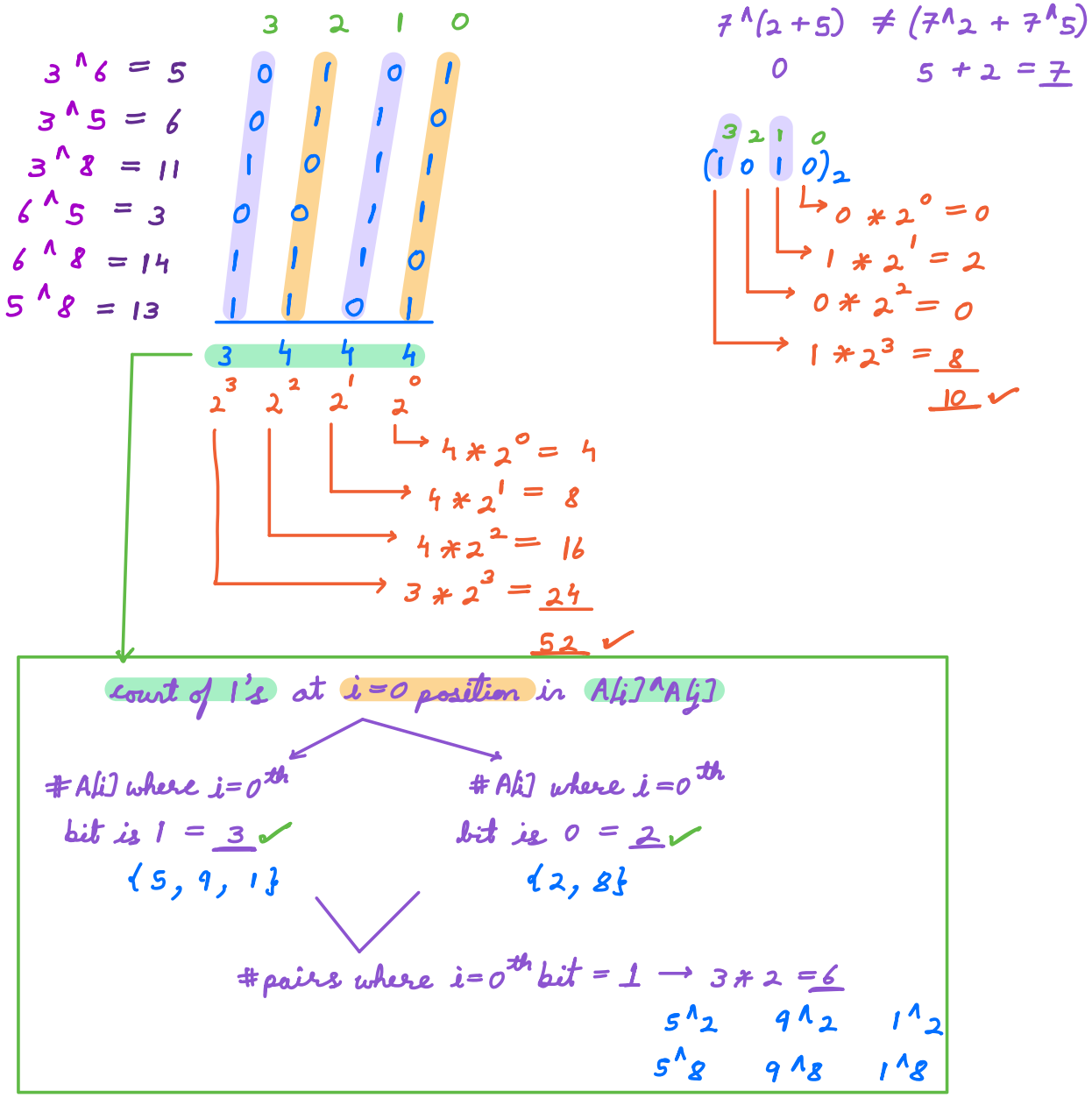
3 → 0 0 1 1
5 → 0 1 0 1
6 → 0 1 1 0

A = [3 6 5 8] (0 1 2 3)

3 ^ 6 = 5
3 ^ 5 = 6
3 ^ 8 = 11
6 ^ 5 = 3
6 ^ 8 = 14
5 ^ 8 = 13
52 (Ans)

8 → 1 0 0 0

Bruteforce → ∀ i, j s.t i < j
calculate A[i] ^ A[j] and
take the sum.
TC = $O(N^2)$ SC = O(1)



count of 1's at i=0 position in A[i]^A[j]

#A[i] where i=0th bit is 1 = 3 ✓ {5, 9, 1}

#A[i] where i=0th bit is 0 = 2 ✓ {2, 8}

#pairs where i=0th bit = 1 → 3 * 2 = 6

5^2 9^2 1^2
5^8 9^8 1^8

A = [3 6 5 8]

	3	2	1	0
3 →	0	0	1	1
6 →	0	1	1	0
5 →	0	1	0	1
8 →	1	0	0	0
#pairs →	3	4	4	4
(A[i]^A[j])	2^3	2^2	2^1	2^0

(2 * 2) 3^6 5^6 3^8 5^8

x^y = a > 0

set bit in a is either set bit in x or y & not both.

↓ ↓ ↓ ↓

24 + 16 + 8 + 4 = 52 (Ans)

a * 5 → 1 step

a + a + a + a + a

```
ans = 0
for b → 0 to 30
    set = 0
    unset = 0
    for i → 0 to (N-1)
        if ((A[i] >> b) & 1 == 1)
            set += 1
        else
            unset += 1
    cnt = set * unset
    ans += cnt * (1 << b)        // 2^b
return ans

// cnt = # pairs with b^th bit = 1.
```

TC = O(N)

SC = O(1)