Question: Given an array where every number occurs thrice except one number. Find it

Ex: 1 12 1 12 4 3 12 1 3 3

Brute Force:

Have 2 loops.

T.C: $O(N^2)$

S.C: $O(1)$

Approach1: Hashmap

→ Construct the freq hashmap

→ Return the element with freq 1

T.C: $O(N)$

S.C: $O(N)$

Approach2: sort

→ Sort the array → $O(N \log N)$

→ Iterate and find the element which occurs once → $O(N)$

1 1 1 3 3 3 6 6 6 7

T.C: $O(N \log N)$

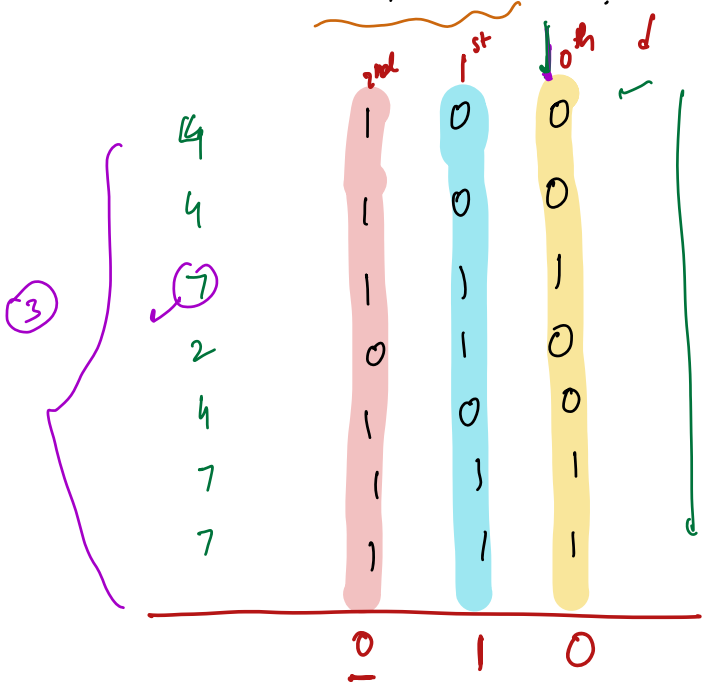
S.C: $O(1)$

Approach:

1 12 1 12 4 3 3 12 1 3 = 1 12 4 3

A = 4 4 7 2 4 7 7

0 1 2 3 4 5 6



32bit

3 0's & 3 1's

(N, i)

(N >> i) & 1

3 1's, 4 0's

3p+1

→ Every number which occurs thrice would contribute 3 0's & 3 1's

1) At every bit, we need count of set bits (t)

2) if(t % 3 == 0) {

} else { // t%3 = 1

ans = ans | (1<<i)

}

```
ans = 0;
for (i = 0; i <= 31; i++) {          // log(Max) + 1
    cnt = 0;
    for (j = 0; j < N; j++) {        // O(N)
        if (A[j] & (1 << i) != 0) {
            cnt++;
        }
    }
    if (cnt % 3 == 1) {
        ans = ans | (1 << i);
    }
}
return ans;
```

TC: $O(N \cdot \log(Max))$

S.C: $O(1)$

Question: All numbers occurs 5 times except one number which occurs either 1 / 2 / 3 / 4 times.

```
if (set_bits % 5 == 0)
    ith bit is 0
else
    ith bit is 1
```

Consider MSB

-10 → 1 3

$\log_2 10 \Rightarrow$

Question: Single Number 3

Every Number occurs even no. of times except 2 numbers. Find out these number

A = 1 2 5 2 5 4 3 4

[1, 3]

Brute force:

Run 2 loops.

T.C: $O(N^2)$

S.C: $O(1)$

Hashing:

→ create a freq hash map

→ Iterate hashmap & find elements with freq 1

T.C: $O(N)$

S.C: $O(N)$

Sorting:

→ Sort the array

→ Iterate over the array to find out elements which occur once

T.C: $O(N \log N)$

S.C: $O(1)$

Efficient Approach

A = 001 (1), 010 (2), 101 (5), 010 (2), 101 (5), 100 (4), 011 (3), 100 (4)

X_r = 1 ^ 3 = 2

001
011
010

Lets assume a, b occur only once

a : _ 1/0 0/1
b : _ 0/1 0/1

a ^ b = 0 1 0

(1) a = 0 0 1
(3) b = 0 1 1
a ^ b = 0 1 0

Y_R = 01101001

A_1 = [1, 5, 5, 4, 4] → 1st bit as 0
A_2 = [2, 2, 3] → 1st bit as 0

Steps

1) Find XOR of entire array (X_r)
2) Find index of any set bit (K)
3) Find XOR of Arr1 & Arr2

X_r = 001000
index: 5 4 3 2 1 0

K = 3

1<<3
A[i] is 1000

// Find XOR of entire array → O(N)

// Find index of a set bit (k) → O(1)

```
Xr1 = 0, Xr2 = 0;
for(i=0; i<N; i++){
    if( A[i] & (1<<k) != 0){
        Xr1 = Xr1 ^ A[i];
    }
    else
        Xr2 = Xr2 ^ A[i];
}
return [ Xr1, Xr2 ];
```

⇒ O(N)

T.C: O(N), S.C: O(1)

→ Find index of any set bit in X_r

X_r =

0	1	1	0	1	0	0	1	0	0
9	8	7	6	5	4	3	2	1	0

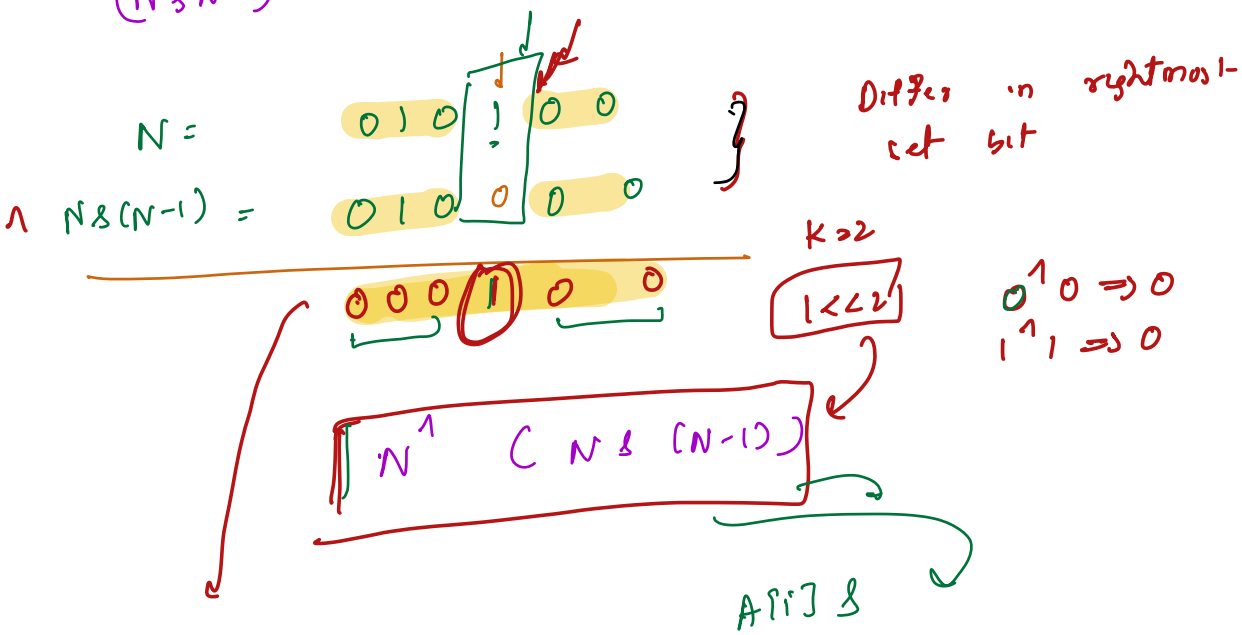
Approach 1:

```
for(i=0; i<= log(Xr); i++){
    if( Xr & (1<<i) != 0){
        return i;
    }
}
```

T.C: O(log(X_r))

Approach2:

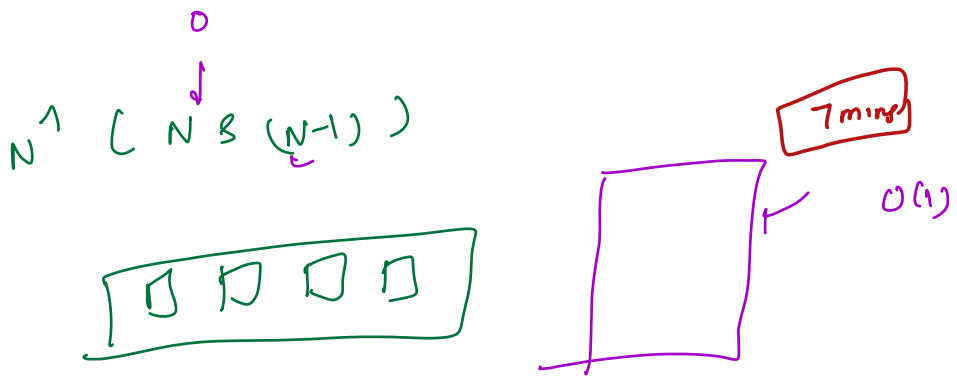
(N & N-1) unsets the rightmost set bit



N = 1001000

N ^ (N & (N-1)) = 1000 as $\log_2$

=> O(1)



Question: Find the 2 missing elements

Array of size N contains elements in the range [1, N+2] except 2 elements. Find these 2 elements (without modifying the array)

A = 3 6 1 4

[2, 5]

A = 1 6 4 7 5

[2, 3]

A = [7 4 6 2 5]

[1, 3]

1) Brute force:

for i in [1, N+2]:

check if exists in array

T.C: $O(N^2)$

S.C: $O(1)$

2) Hashset:

→ Insert elements into hashset

→ for i in [1, N+2], check if it exists

T.C: $O(N)$ S.C: $O(N)$

3) Sort

→ Sort the array

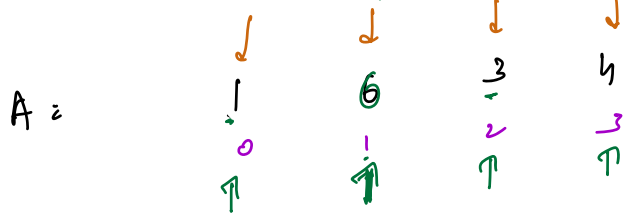
→ Iterate and find the elements which are missing

T.C: O(N log N)

S.C: O(N)

4) Map element to (N+1) and (N+2) th index

A :



N = 4

[1 - N]

0 → 1

1 → 2

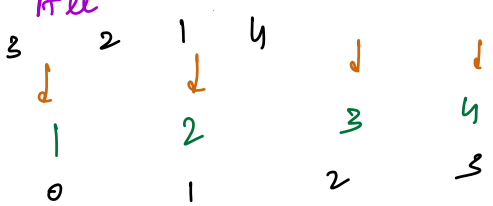
⋮

N-1 → N

2 is missing

Case1 : All elements are in the position

A :



Ans = [N+1, N+2]

Case2 : 2 elements are not in position

A : 6 3 5 2

A = 6 2 3 5

0 1 2 3

1 is missing

4 is missing

Cases:

A = 2 6 3 1

A = 1 2 3 6

0 1 2 3

→ N+1 / N+2

→ 4 is missing

→ N+1 (5) is other missing number

T.C : O(N)

S.C : O(N)

5) Approach 5 :

A = 3 6 1 4 1 2 3 4 5 6

0 1 2 3

N = 4

[1, N+2]

a ^ b

T.C : O(N)

S.C : O(1)

6) Approach 6 :

[1,6] : $\frac{6 \times 7}{2}$,

N = 4

A = 3 6 ! 4

sum(1 .... N+2) =>

$\frac{(N+2)(N+3)}{2} = \frac{6 \times 7}{2} = 21$

a + b = 21 − sum(A)

= 21 − 14

a ↑ 7 → a + b = 7 — (1)

$a \uparrow b = 2 \uparrow 5 = 7$

2 ← 0 1 0

5 ← 1 0 1

1 1 1

$a \uparrow b = 7$ — (2)

XOR with a or b in (1) on both sides

$a \uparrow (a+b) = a \uparrow 7$

$= a \uparrow a + a \uparrow b = a \uparrow 7$

$0 + 7 = a \uparrow 7$

$7 = a \uparrow 7$

a = 0

$b \uparrow (a+c) \neq b \uparrow a + b \uparrow c$

Might not be true

N = 4

A = 1 2 6 5

{a = 3
 b = 4
}

0 1 1
1 0 0
1 1 1

1 + 4 + 36 + 25 =

sum(1, N+2) = 21

sum(arr) = 14

a + b = 7 — (1)

a ^ b = 7 — (2)

$a \wedge (a+b) = a \wedge 7$

Distributive property does not hold

$a \wedge a + a \wedge b = a \wedge 7$

0

$a \wedge b = a \wedge 7$

$7 = a \wedge 7$

$a = 0$ ✗

$1^2 + 2^2 + 3^2 + \cdots N^2 = \frac{N(N+1)(2N+1)}{6}$

$1^2 + 2^2 + 3^2 + \cdots 6^2$

sum(squares of array) = 66

$= \frac{6(7)(13)}{6}$

$= 91$

$a^2 + b^2 = 91 - 66 = 25$

$$a+b=7$$
$$a^2+b^2=25$$

$(a+b)^2 = a^2+b^2+2ab$

$49 = 25+2ab$

$2ab = 24$

$$ab = 12$$

$(a-b)^2 = a^2+b^2-2ab$

$= 25-2(12)$

$(a-b)^2 = 25-24$

$(a-b)^2 = 1$

$a-b = \pm 1$

$a+b = 7$

$a-b = 1$

$2a = 8 \Rightarrow$

$$a = 4$$
$$b = 3$$

10^6

$[1, N+2]$

$N = 10^6$

$(10^6)^2 = 10^{12}$

$10^{12} \times 10^6 = 10^{18}$

10^{18} ?

$a \uparrow (b \uparrow c) = (a \uparrow b) \uparrow c$

$a \uparrow (b+c) \neq a \uparrow b + a \uparrow c$ ✓

Monday Moving IST
Sunday Night EST

3-4 problems

$(a+b)\%M = (a\%M + b\%M)\%M$

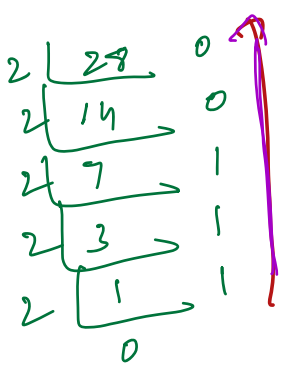
$(a*b)\%M = (a\%b * b\%M)\%M$

$(a-b)\%M = (a\%M - b\%M + M)\%M$

N

28

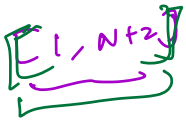
Arrays



11100

$\log_2(28)$

$N \to \frac{N}{2} \to \frac{N}{4} \to \frac{N}{8} \cdots \to 1 \Rightarrow \log_2 N$


[1, N+2]


a+b