Single Number 2

02 December 2023 07:25 PM

Problem Description

Given an array of integers, every element appears thrice except for one, which occurs once.

Find that element that does not appear thrice.

NOTE: Your algorithm should have a linear runtime complexity.

Could you implement it without using extra memory?

From <https://www.scaler.com/academy/mentee-dashboard/class/90013/assignment/problems/195>

Problem Constraints

- 2 <= A <= $5*10^6$
- 0 <= A <= INTMAX

Input Format

First and only argument of input contains an integer array A.

Output Format

Return a single integer.

Example Input

Input 1:
A = [1, 2, 4, 3, 3, 2, 2, 3, 1, 1]
Input 2:
A = [0, 0, 0, 1]

Example Output

Output 1:
4
Output 2:
1

[1, 2, 4, 3, 3, 2, 2, 3, 1, 1]

i) BruteForce approach :-

iterate over each element & find its duplicate by iterating over array again.

But T.C. $O(n^2)$ which is not acceptable.

ii)

other approach would be using anothe Data Structure array, Hashmap,

But this is also not acceptable since we can not use any extra memory.

iii) Sort the array & Check its frequency

But T.C. O(n logn)

iv) Bit Manipulation :

Approach :-

Approach

If no of 1's at ith bit is not divisible by 3
⇒ unique number is Contributing a 1 at that bit.

[1, 2, 4, 3, 3, 2, 2, 3, 1, 1]

1	0 0 0 0 0 0 0 1
2	0 0 0 0 0 0 1 0
4	0 0 0 0 0 1 0 0
3	0 0 0 0 0 0 1 1
3	0 0 0 0 0 0 1 1
2	0 0 0 0 0 0 1 0
2	0 0 0 0 0 0 1 0
3	0 0 0 0 0 0 1 1
1	0 0 0 0 0 0 0 1
1	0 0 0 0 0 0 0 1
	0 0 0 0 0 0 1 6 6

divisible by 3

6/3 = 0
6/3 = 0
1/3 = 1

Set the bit.

0 1 1<<2

0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0

0 0 0 0 0 1 0 0

Pseudocode :-

```
ins ans =0;
for (int i=0 to 32) {
```

```
checkbit(int num, index) {
return
```

```
        int Cnt = 0 ;
    for(int j=0 ; j< A.size(); j++) {
          if (Checkbit(A[j] , i) == true) {
               Cnt ++ ;
          }
    }
    If (Cnt%3 != 0) {   //set the ith bit
        ans = ans | 1<<i ;
    }
  }
}
```

```
return   num & 1<<Indx ==
                  1<<Indx
}
```