Leetcode : Single Number

Friday, April 29, 2022 9:16 PM

Given a **non-empty** array of integers nums, every element appears *twice* except for one. Find that single one.
You must implement a solution with a linear runtime complexity and use only constant extra space.

From <https://leetcode.com/problems/single-number/>

Approach: i) While each Number have exactly 2 occurance except the one. So if we calculate the Xor of Whole array , we will get desired output.
(ii) Because we know that XOR of two same Number is '0'.As a result those number have Frequency 2, will be xor of these is Zero. So only the single occurance based number exist After the operation.

Ex: Suppose We have an array → 4, 1, 2, 1, 2;

Step 1: x = 4; → x ^= arr[1] = 5.
Step 2: x = 5; → x ^= arr[2] = 7
Step 3: x = 7; → x ^= arr[3] = 6
Step 4: x = 6; → x ^= arr[4] = 4

Ans

C++ :

```
class Solution {
public:
    int singleNumber(vector<int>& nums) {
        int x = 0;
        for(int i=0;i<nums.size();i++)
        {
            x^=nums[i];
        }
        return x;
    }
};
```

T.C → O(n);
S.C → O(1);