136. Single Number

Companies

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.

**Example 1:**

**Input:** nums = [2,2,1]

**Output:** 1

**Example 2:**

**Input:** nums = [4,1,2,1,2]

**Output:** 4

**Example 3:**

**Input:** nums = [1]

**Output:** 1

**Constraints:**

* 1 <= nums.length <= 3 \* 104
* -3 \* 104 <= nums[i] <= 3 \* 104
* Each element in the array appears twice except for one element which appears only once.

Solution

public class Solution {

public int SingleNumber(int[] nums) {

int ans = nums[0];

for(int i = 0; i < nums.Length; i++){

int check = nums[i];

int count = 0;

for(int j = 0; j< nums.Length; j++){

if(check == nums[j]){

count += 1;

}

}

if(count == 1){

ans = check;

}

count = 0;

}

return ans;

}

}