

Description | Accepted | Editorial | Solutions | Submissions

260. Single Number III

Solved

Medium

Topics

Companies

Given an integer array `nums`, in which exactly two elements appear only once and all the other elements appear exactly twice. Find the two elements that appear only once. You can return the answer in **any order**.

You must write an algorithm that runs in linear runtime complexity and uses only constant extra space.

Example 1:

Input: `nums = [1,2,1,3,2,5]`

Output: `[3,5]`

Explanation: `[5, 3]` is also a valid answer.

Example 2:

Input: `nums = [-1,0]`

Output: `[-1,0]`

Example 3:

Input: `nums = [0,1]`

Output: `[1,0]`

Constraints:

6.3K | 132 | ?

Problem List | Run | Submit | Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted

ZhukovIlya submitted at Jul 07, 2024 15:46

Editorial | Solution

Runtime: 112 ms | Beats: 73.72% | Memory: 48.09 MB | Beats: 77.17%

Analyze Complexity

10%

5%

0%

80ms 107ms 125ms 143ms 161ms 180ms

Code | C#

```
public class Solution {
    public int[] SingleNumber(int[] nums) {
        // 1. Найти XOR всех элементов
        int xor = 0;
        foreach (int num in nums) {
            xor ^= num;
        }

        // 2. Найти бит, который отличается у двух уникальных чисел
        int diffBit = xor & -xor; // находим самый младший бит, который отличается
    }
}
```

Testcase | Test Result

Accepted Runtime: 97 ms

Case 1 | Case 2 | Case 3

Input

nums =

[1,2,1,3,2,5]

Output

[5,3]

Expected

Код:

```
public class Solution
{
    public int[] SingleNumber(int[] nums)
```

```

{
    // 1. Найти XOR всех элементов
    int xor = 0;
    foreach (int num in nums)
    {
        xor ^= num;
    }

    // 2. Найти бит, который отличается у двух уникальных чисел
    int diffBit = xor & -xor; // находим самый младший бит, который отличается

    int[] result = new int[2];
    // 3. Разделить все числа на две группы и найти уникальные числа
    foreach (int num in nums)
    {
        if ((num & diffBit) == 0)
        {
            result[0] ^= num;
        }
        else
        {
            result[1] ^= num;
        }
    }

    return result;
}
}

```