Description | Accepted | Editorial | Solutions | Submissions

922. Sort Array By Parity II

Solved

Easy | Topics | Companies

Given an array of integers `nums`, half of the integers in `nums` are **odd**, and the other half are **even**.

Sort the array so that whenever `nums[i]` is odd, `i` is **odd**, and whenever `nums[i]` is even, `i` is **even**.

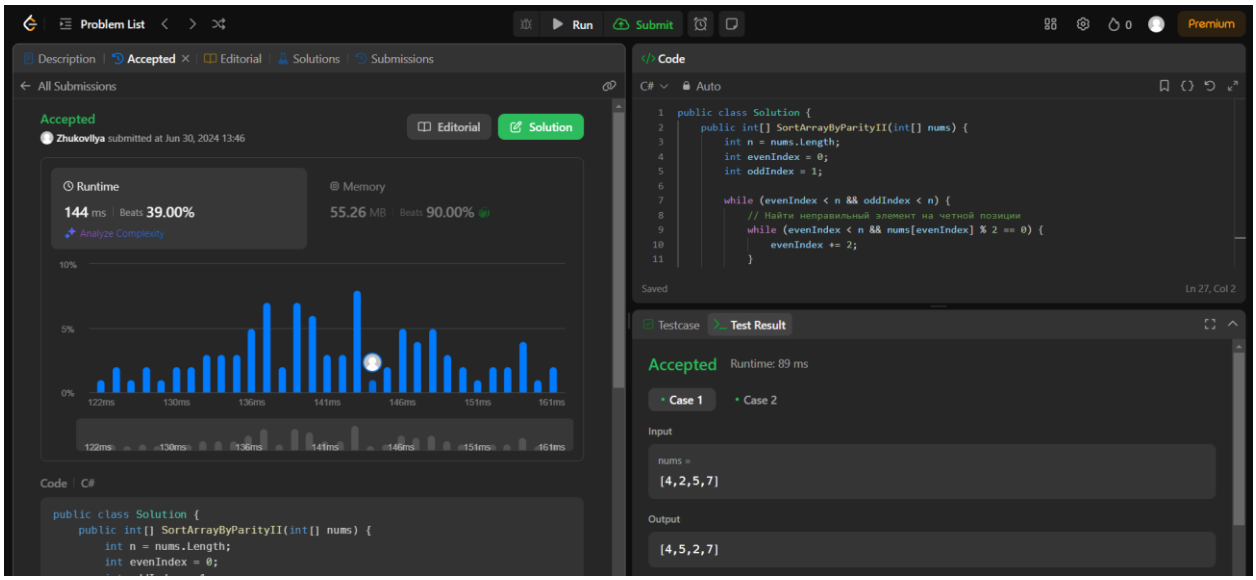
Return *any answer array that satisfies this condition*.

Example 1:

```
Input: nums = [4,2,5,7]
Output: [4,5,2,7]
Explanation: [4,7,2,5], [2,5,4,7], [2,7,4,5] would also have been accepted.
```

Example 2:

```
Input: nums = [2,3]
Output: [2,3]
```



Код:

```
public class Solution
{
    public int[] SortArrayByParityII(int[] nums)
    {
        int n = nums.Length;
        int evenIndex = 0;
        int oddIndex = 1;

        while (evenIndex < n && oddIndex < n)
        {
            // Найти неправильный элемент на четной позиции
            while (evenIndex < n && nums[evenIndex] % 2 == 0)
            {
```

```
                evenIndex += 2;
            }
            // Найти неправильный элемент на нечетной позиции
            while (oddIndex < n && nums[oddIndex] % 2 == 1)
            {
                oddIndex += 2;
            }

            // Если оба индекса в пределах массива, обменять элементы
            if (evenIndex < n && oddIndex < n)
            {
                int temp = nums[evenIndex];
                nums[evenIndex] = nums[oddIndex];
                nums[oddIndex] = temp;
            }
        }

        return nums;
    }
}
```