Problem 922: Sort Array By Parity II

Problem Information

Difficulty: Easy
Acceptance Rate: 71.13%
Paid Only: No
Tags: Array, Two Pointers, Sorting

Problem Description

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]

Constraints:

* `2 <= nums.length <= 2 * 104` * `nums.length` is even. * Half of the integers in `nums` are even. * `0 <= nums[i] <= 1000`

Follow Up: Could you solve it in-place?

Code Snippets

C++:

```
class Solution {
public:
vector<int> sortArrayByParityII(vector<int>& nums) {

}
};
```

Java:

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

}
}
```

Python3:

```
class Solution:
def sortArrayByParityII(self, nums: List[int]) -> List[int]:
```