922. Sort Array By Parity II

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?

```
def sortArrayByParityII(self, nums: List[int]) -> List[int]:
#Solution to the follow-up
        i = 0
        j = 0
        while i<len(nums):
            if nums[i]%2!=0:
                i = i+1
            else:
                nums[i],nums[j] = nums[j],nums[i]
                i = i+1
                j = j+1
        i = 1
        mid = len(nums)//2
        if mid%2==0:
            j = mid
        else:
            j = mid+1
        while i<len(nums)//2:
            nums[i],nums[j] = nums[j],nums[i]
            i = i+2
            j = j+2
        return nums
    #     res = [0]*len(nums)
    #     i = 0
    #     j = 1
    #     k = 0
    #     while k<len(nums):
    #         if nums[k]%2==0:
    #             res[i]=nums[k]
    #             i = i+2
    #         else:
    #             res[j]= nums[k]
    #             j = j+2
    #         k = k +1
    #     return res
```