905. Sort Array By Parity

02 May 2022 05:03 PM

Given an integer array `nums`, move all the even integers at the beginning of the array followed by all the odd integers.

Return ***any array*** *that satisfies this condition.*

Example 1:

```
Input: nums = [3,1,2,4]
Output: [2,4,3,1]
Explanation: The outputs [4,2,3,1], [2,4,1,3], and
[4,2,1,3] would also be accepted.
```

Example 2:

```
Input: nums = [0]
Output: [0]
```

nums = [3, 1, 2, 4]

output

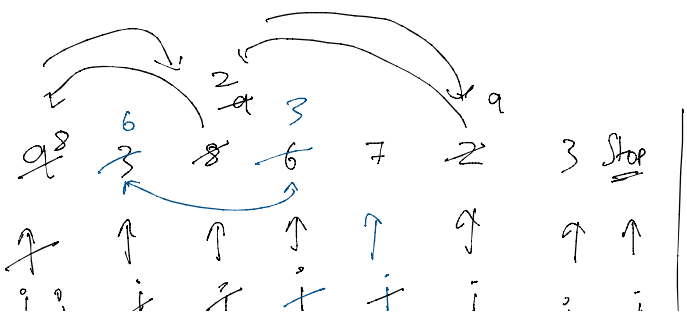
[2, 4, 3, 1]

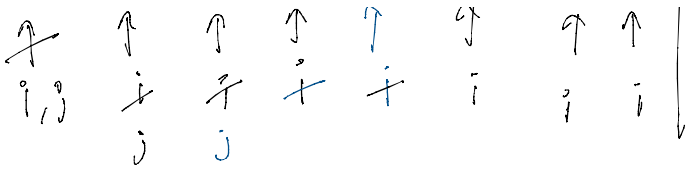
3, 1, 2, 4

↑ ↑

l r

```
while (l < r) {
    if (A[l] % 2 != 0)
        swap [A[l], A[r]];
        --r;
    else {
        ++l
    }
return A
```

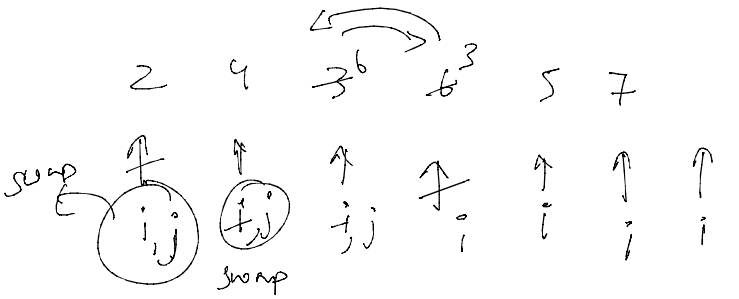




i = 0

j = 0

* keep increment j
* when i == even , swap (i, j) and i++, j++ [Peploding]



* when i == even , i++, j++ and swap (i, j)

when odd just i++

```
class Solution {
    public int[] sortArrayByParity(int[] nums) {
        int i = 0; // first unsolved
        int j = 0; // first odd

        while(i < nums.length){
            if(nums[i] % 2 == 0){
                //even -> swap(i,j), i++, j++
                int temp = nums[i];
                nums[i] = nums[j];
                nums[j] = temp;

                i++;
                j++;
            } else {
                // odd -> i++
                i++;
            }
        }
        return nums;
    }
}
```