

JavaScript Sort Array by Parity

Challenge

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.

1st Example

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.



2nd Example

Input: `nums = [0]`

Output: `[0]`




Constraints

- `1 <= nums.length <= 5000`
- `0 <= nums[i] <= 5000`

Solution

```
const sortByParity = (nums) => {  
  let oddIndex = 0;  
  
  for (let i = 0; i < nums.length; i++) {  
    if (nums[i] % 2 !== 0) {  
      [nums[i], nums[oddIndex]] = [nums[oddIndex],  
                                   nums[i]];  
      oddIndex++;  
    }  
  }  
  
  return nums;  
};
```



Explanation

I've coded a function called `sortByParity` that takes an array of numbers, `nums`, as input. The purpose of this function is to rearrange the elements of the input array such that all even numbers appear before odd numbers, while maintaining the relative order of even and odd numbers within their respective groups.

The function begins by initializing a variable called `oddIndex` with a value of `0`. This variable will keep track of the index where the next odd number should be placed in the array.

Next, a `for` loop is used to iterate through each element of the input array. Within the loop, an `if` statement checks if the current

number `(nums[i])` is even by checking if the remainder of dividing it by `2` is `0`.

If the number is even, a destructuring assignment is used to swap the current number with the number at the `oddIndex`. This effectively moves the even number to its correct position in the array, ensuring that all even numbers appear before odd numbers.

After the swap, the `oddIndex` is incremented by `1` to prepare for the next odd number that will be encountered.

Once the loop finishes, the modified array is returned as the output of the function.

In summary, this function rearranges the elements of the input array such that all even numbers come before odd numbers, while preserving the relative order of even and odd numbers within their respective groups.