

JavaScript Rotate Array

Challenge

Given an integer array `nums`, rotate the array to the right by `k` steps, where `k` is non-negative.

1st Example

Input: `nums = [1,2,3,4,5,6,7]`, `k = 3`

Output: `[5,6,7,1,2,3,4]`

Explanation: rotate 1 steps to the right: `[7,1,2,3,4,5,6]`
rotate 2 steps to the right: `[6,7,1,2,3,4,5]`
rotate 3 steps to the right: `[5,6,7,1,2,3,4]`

2nd Example

Input: `nums = [-1,-100,3,99]`, `k = 2`

Output: `[3,99,-1,-100]`

Explanation: rotate 1 steps to the right: `[99,-1,-100,3]`
rotate 2 steps to the right: `[3,99,-1,-100]`

Constraints

- $1 \leq \text{nums.length} \leq 10^5$
- $-2^{31} \leq \text{nums}[i] \leq 2^{31} - 1$
- $0 \leq k \leq 10^5$

Solution

```
const rotate = (nums, k) => {  
  for (let i = nums.length - 1; i >= 0; i--) {  
    nums[i + k] = nums[i];  
  }  
  
  for (let j = k - 1; j >= 0; j--) {  
    nums[j] = nums.pop();  
  }  
};
```



Explanation

I've written a function called `rotate` which takes in an array `nums` and a number `k` as parameters. The function is used to rotate the elements of the array `nums` to the right by `k` positions.

This function consists of two `for` loops. The first `for` loop starts from the last index of the array `nums` and iterates backwards until it reaches the first index. Inside this loop, each element at index `i` is assigned to the element at index `i + k` in the array `nums`. This effectively shifts the elements to the right by `k` positions.

After the first `for` loop, the elements of the array `nums` have been shifted to the right, but the first `k` elements are duplicates of the original elements.

The second `for` loop starts from `k - 1` and iterates backwards until it reaches `0`. Inside this loop, the last element of the array `nums` is removed using the `pop` method, and the removed element is assigned to the current element at index `j` in the array

`nums`. This effectively replaces the duplicate elements at the beginning of the array with the original elements.

Once the second `for` loop completes, the array `nums` will contain the elements rotated to the right by `k` positions.

It's important to note that this function modifies the original array `nums` in place and does not return a new array.