Array #11

Leetcode #283

Move Zeroes

https://leetcode.com/problems/move-zeroes/

Given an array of integers nums, move all zeroes to the end of it while maintaining the relative order of the non-zero elements

NOTE: You must do this in-place without making a copy of the array

Example 1:
Input: nums = [0, 1, 0, 3, 12]
Output: [1, 3, 12, 0, 0]

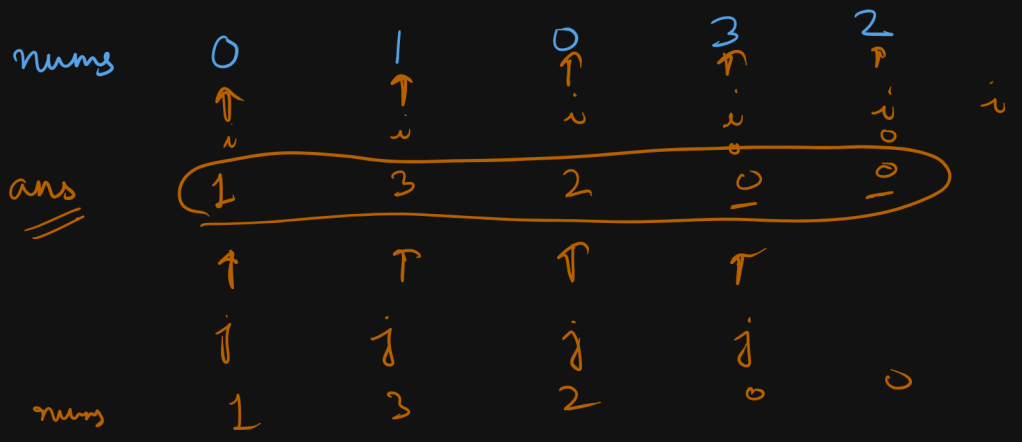
1 3 12 0 0

Example 2:
Input: nums = [0]
Output: [0]

Constraints:
1 <= nums.length <= 10^4
-2^31 <= arr[i] <= 2^31-1

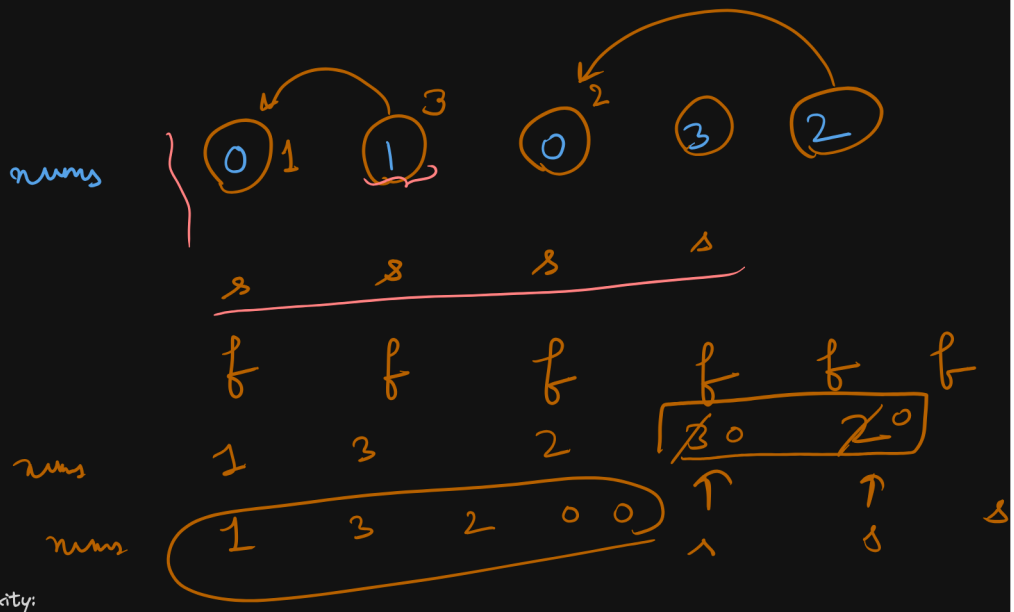
Companies:
Meta, Amazon, Yandex, Microsoft, Google, Adobe, etc

Approach 1:



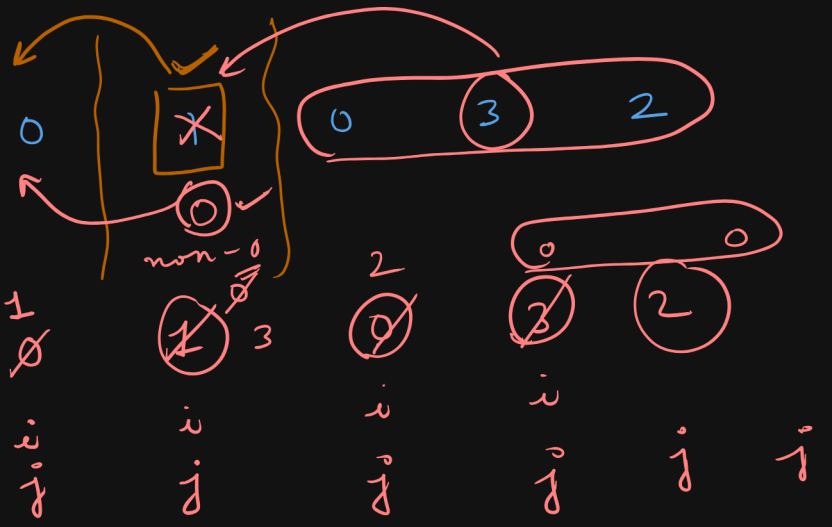
Time complexity:
O(N)
Space Complexity:
O(N)

Approach 2:



Time complexity:
O(N)
Space Complexity:
O(1)

Approach 3:



Time complexity:
O(N)
Space Complexity:
O(1)

```
class Solution {
    public void moveZeroes(int[] nums) {
        // approach 1
```

```
        int[] res = new int[nums.length];
        int j = 0;

        for (int i = 0; i < nums.length; i++) {
            if (nums[i] != 0) {
                res[j] = nums[i];
                j++;
            }
        }

        while (j < nums.length) {
            res[j] = 0;
            j++;
        }

        for (int i = 0; i < nums.length; i++) {
            nums[i] = res[i];
        }
    }
}


class Solution {
    public void moveZeroes(int[] nums) {
        // approach 2
        int slow = 0,
            fast = 0;

        while (fast < nums.length) {
            if (nums[fast] != 0) {
                nums[slow] = nums[fast];
                slow++;
            }

            fast++;
        }

        while (slow < nums.length) {
            nums[slow] = 0;
            slow++;
        }
    }
}


class Solution {
    public void moveZeroes(int[] nums) {
        // approach 2
        int slow = 0,
            fast = 0;

        while (fast < nums.length) {
            if (nums[fast] != 0) {
                int temp = nums[fast];
                nums[fast] = nums[slow];
                nums[slow] = temp;
                slow++;
            }
```


fast++;