

Given an integer array nums, move all 0's to the end of it while maintaining the relative order of the non-zero elements.

Note that 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]

Example 2:

Input: nums = [0]

Output: [0]

Constraints:

a. $1 \leq \text{nums.length} \leq 10^4$

b. $-2^{31} \leq \text{nums}[i] \leq 2^{31} - 1$

```
class Solution {
    public void moveZeroes(int[] nums) {
        if(nums.length>1){
            int i=0;
            int j=1;
            while(i<nums.length && j<nums.length){
                if(nums[i]==0 && nums[j]!=0){
                    int temp=nums[i];
                    nums[i]=nums[j];
                    nums[j]=temp;
                    i++;
                    j++;
                }else if(nums[i]==0 && nums[j]==0){
                    j++;
                }else if(nums[i]!=0 && nums[j]!=0){
                    i++;
                    j++;
                }else if(nums[i]!=0 && nums[j]==0){
                    i++;
                    j++;
                }
            }
        }
    }
}
```

```

        }
    }

}

```

First Unique Character in a String

Given a string *s*, find the first non-repeating character in it and return its index. If it does not exist, return -1.

Example 1:

Input: *s* = "leetcode"

Output: 0

Example 2:

Input: *s* = "loveleetcode"

Output: 2

Example 3:

Input: *s* = "aabb"

Output: -1

Constraints:

- $1 \leq s.length \leq 10^5$
- s* consists of only lowercase English letters.

Note: Create a GitHub file for the solution and add the file link the the answer section below.

/ Online Java Compiler and Editor */*

```

public class HelloWorld{

    public static int firstUniqChar(String s) {
        int[] frequency = new int[26]; // Frequency map for lowercase English letters

        // Update frequency map
        for (char ch : s.toCharArray()) {
            frequency[ch - 'a']++;
        }
    }
}

```

```
// Find the first non-repeating character
for (int i = 0; i < s.length(); i++) {
    if (frequency[s.charAt(i) - 'a'] == 1) {
        return i;
    }
}

return -1; // No non-repeating character found
}

public static void main(String[] args)
{
    System.out.print(firstUniqChar("aabb"));
}
}
```