LeetCode 283: Move Zeroes

Zeyong Jin*

School of Computing Sciences, Simon Fraser University

September 4, 2021

Here I presented a pseudo-code to solve the 283rd question of LeetCode: Move Zeroes.

The description of this question is that given an integer array `nums`, move all 0's to the end of it while maintaining the relative order of the non-zero elements.

One **note** of this question is that you must do this in-place without making a copy of the array.

The constranits of this question are as follows:

(a) $1 \leq$ `nums`.length $\leq 10^4$.

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

One **follow up** question is that whether you could minimize the total number of operations done.

For more information about this question, click on the following link: Move Zeroes or go to the next url: https://leetcode.com/problems/move-zeroes/.

The pseudo-code is in the following page.

The time complexity of this algorithm is $O(n)$.

The space complexity of this algorithm is $O(1)$.

*zeyongj@gmail.com

Algorithm 1: Move Zeroes

Input: int& nums
Output: none

```
1  cursor ← 0;
2  size ← length(nums);
3  i ← 0;
4  while i < size do
5  |  if nums[i] != 0 then
6  |  |  nums[cursor] = nums[i];
7  |  |  cursor++ ;    /* Cursor move to the next index waiting
   |  |   to be exchanged.  */
8  |  else
9  |  |  continue ;
10 |  end
11 end
12 i ← cursor ;     /* Cursor now at the index next to the final
    position of non-zero value, i.e.  cursor now points at the
    index of the first zero value.  */
13 while i < size do
14 |  nums[i] = 0 ;      /* Starting from the cursor, all values
   |   would be 0s.  */
15 end
```