SlidesPractice_L1_Array_Diagram

3/31/17 11:58 AM

1. Moving zeros

1, 0, 2, 0, 3 ⇒ 1, 2, 3, 0, 0

✓ x ✓ x ✓

1, 0, 2, 0, 3

two pointers

i (i: iterate the input array)

↓

1, 0, 2, 0, 3

↑

j (j: represents the index of result array)

if (nums[i] != 0) {

① let nums[j] = nums[i]; (nums[j] = nums[i];)

② move j forward (j++;)

(1) i (nums[i] = 1) → i (nums[i] = 0)

1, 0, 2, 0, 3 ⇒ 1, 0, 2, 0, 3

j → j

(3) i (nums[i] = 2) → i (nums[i] = 0)

1, 0, 2, 0, 3 ⇒ 1, 2, 2, 0, 3

j → j

(5) i (nums[i] = 3) → i >= nums.length

1, 2, 2, 0, 3 ⇒ 1, 2, 3, 0, 3

j → j

} else {

③ only move i forward until find non-zero element. (using a for loop)

(2) i (nums[i] = 0) → i (nums[i] = 2)

1, 0, 2, 0, 3 ⇒ 1, 0, 2, 0, 3

j → j

(4) i (nums[i] = 0) → i (nums[i] = 3)

1, 2, 2, 0, 3 ⇒ 1, 2, 2, 0, 3

j → j

}

When i >= nums.length, let all elements in [nums[j], nums[nums.length - 1]] = 0;

i >= nums.length

1, 2, 3, 0, 3 ⇒ 1, 2, 3, 0, 0

j → j

2. Rotate Array by K Steps

3 step reversing (k=3)

Original Array: [1, 2, 3, 4, 5, 6, 7] (n=7)

① Reverse All: [7, 6, 5, 4, 3, 2, 1]

② Reverse first k: [5, 6, 7, 4, 3, 2, 1]

③ Reverse last n-k: [5, 6, 7, 1, 2, 3, 4]