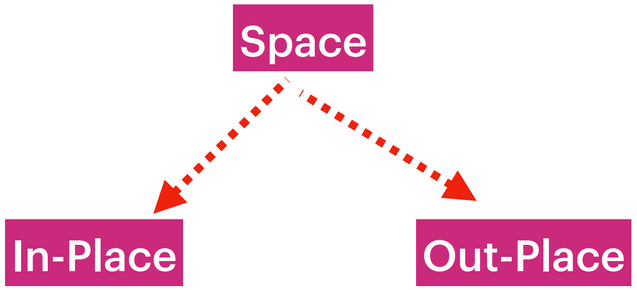

Space
In-Place
Out-Place

287. Find the Duplicate Number

Medium 12229 1366 Add to List Share

Given an array of integers `nums` containing `n + 1` integers where each integer is in the range `[1, n]` inclusive.

There is only **one repeated number** in `nums`, return *this repeated number*.

You must solve the problem **without** modifying the array `nums` and uses only constant extra space.

Example 1:

```
Input: nums = [1,3,4,2,2]
Output: 2
```

Example 2:

```
Input: nums = [3,1,3,4,2]
Output: 3
```

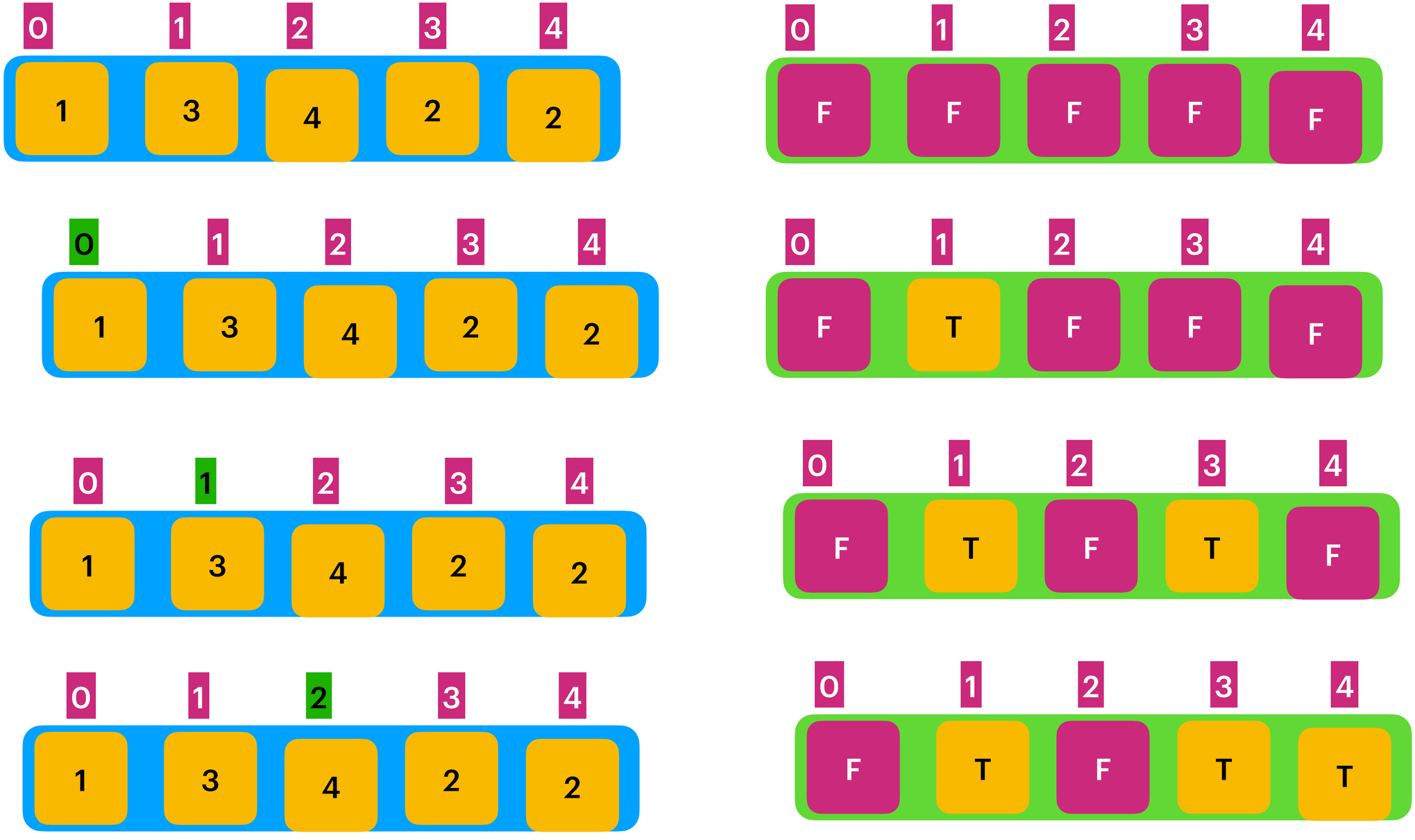
Constraints:

- `1 <= n <= 10^5`
- `nums.length == n + 1`
- `1 <= nums[i] <= n`
- All the integers in `nums` appear only **once** except for **precisely one integer** which appears **two or more** times.

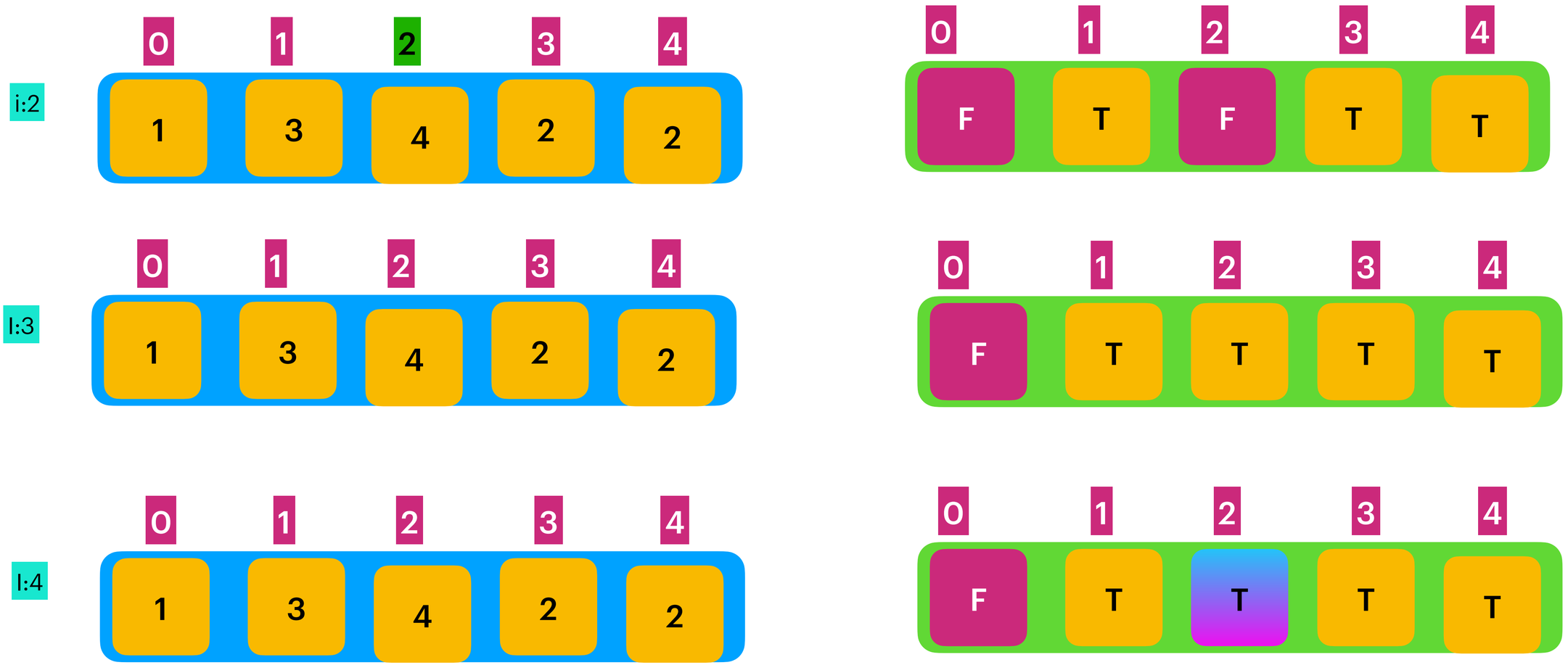
Follow up:

- How can we prove that at least one duplicate number must exist in `nums`?
- Can you solve the problem in linear runtime complexity?

OutPlace Algo



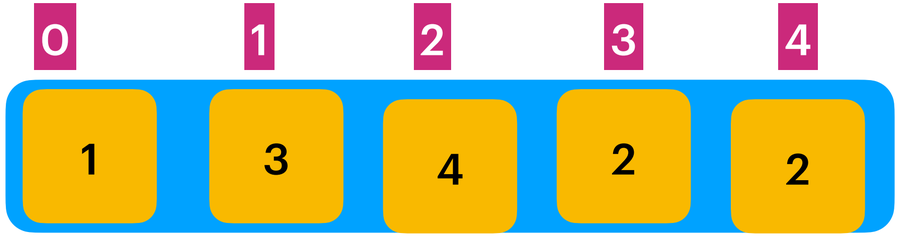
OutPlace Algo



Time Complexity : O(n)

Excepted Output : 2

Space Complexity : O(n)

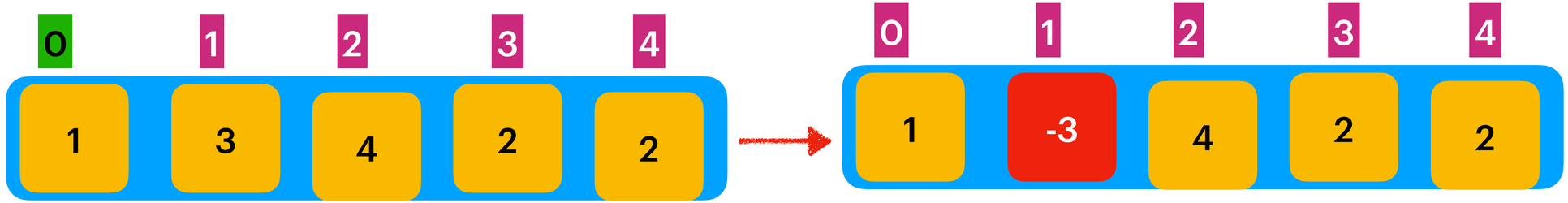


InPlace Algo

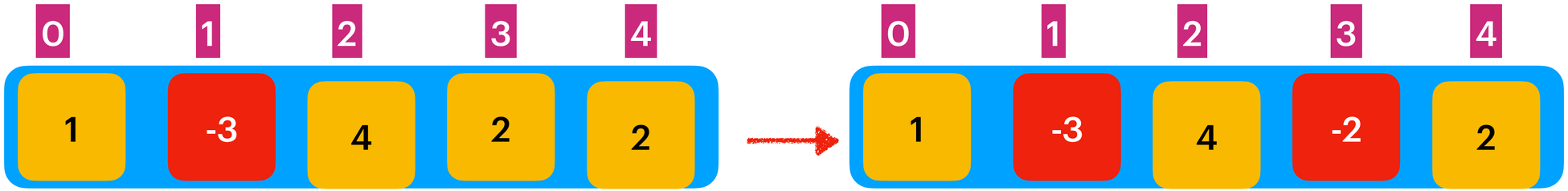
Input

Output

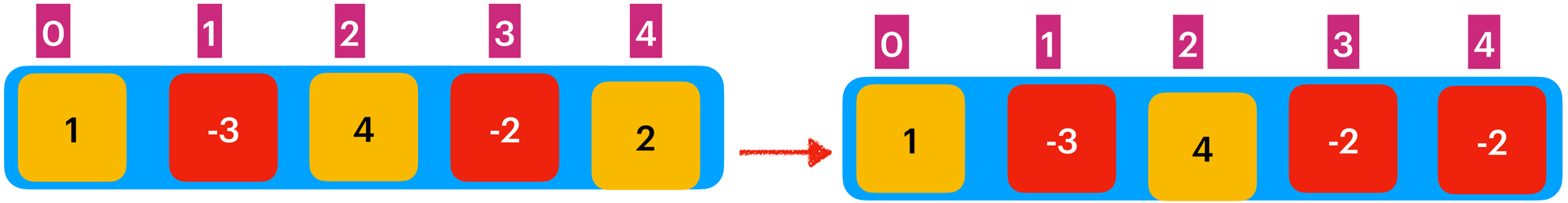
i=0



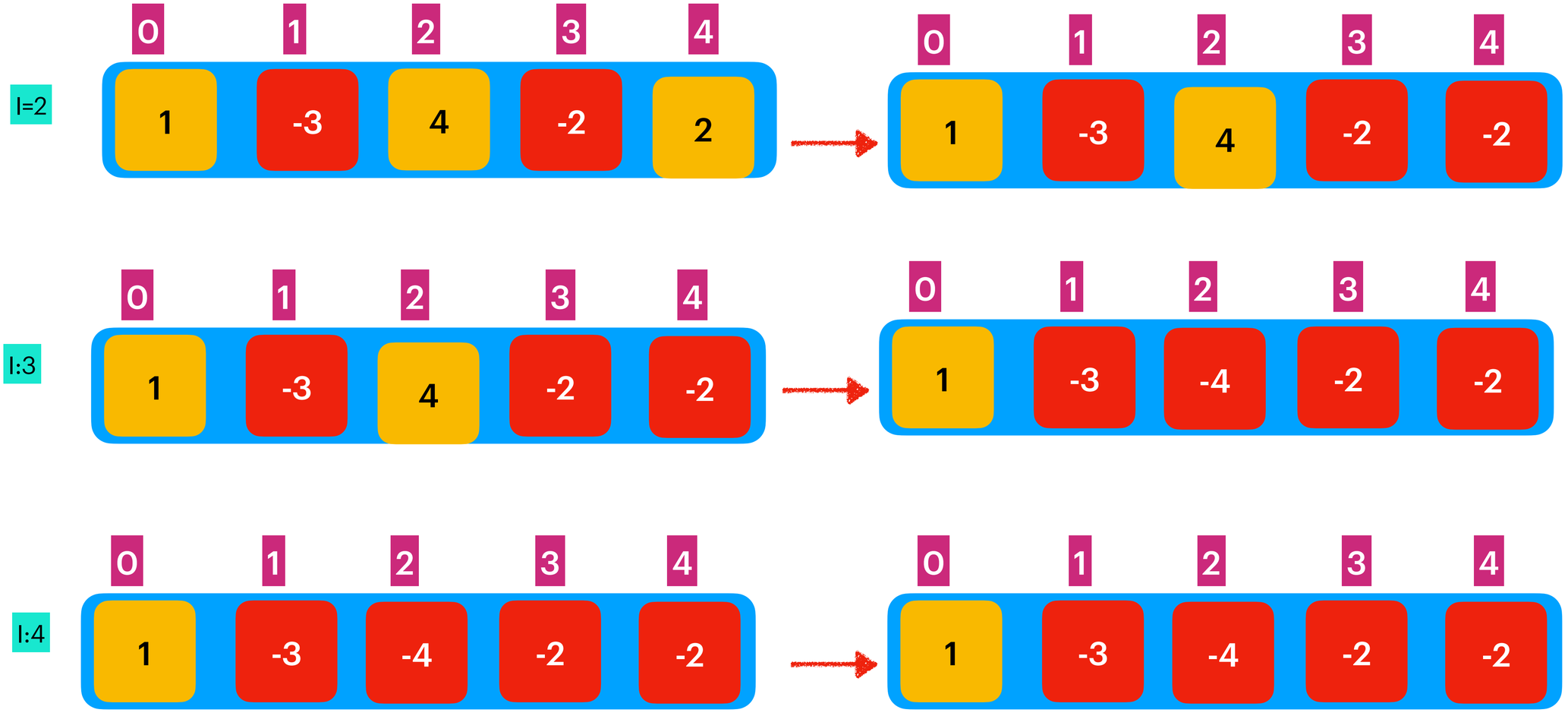
I=1



I=2



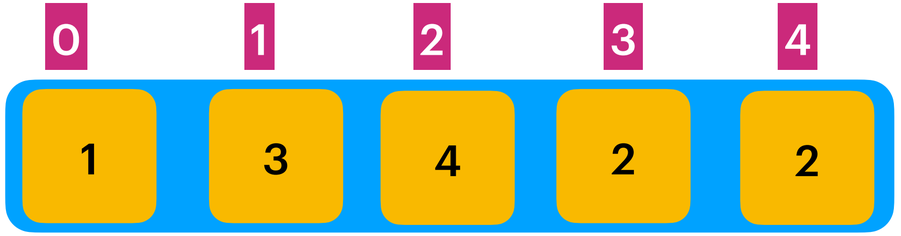
In-Place Algo



Time Complexity : O(n)

Excepted Output : 2

Space Complexity : O(1)



41. First Missing Positive

Hard 8998 1283 Add to List Share

Given an unsorted integer array `nums`, return the smallest missing positive integer.

You must implement an algorithm that runs in `O(n)` time and uses constant extra space.

Example 1:

```
Input: nums = [1,2,0]
Output: 3
```

Example 2:

```
Input: nums = [3,4,-1,1]
Output: 2
```

Example 3:

```
Input: nums = [7,8,9,11,12]
Output: 1
```

Constraints:

- $1 \le nums.length \le 5 * 10^5$
- $-2^{31} \le nums[i] \le 2^{31} - 1$