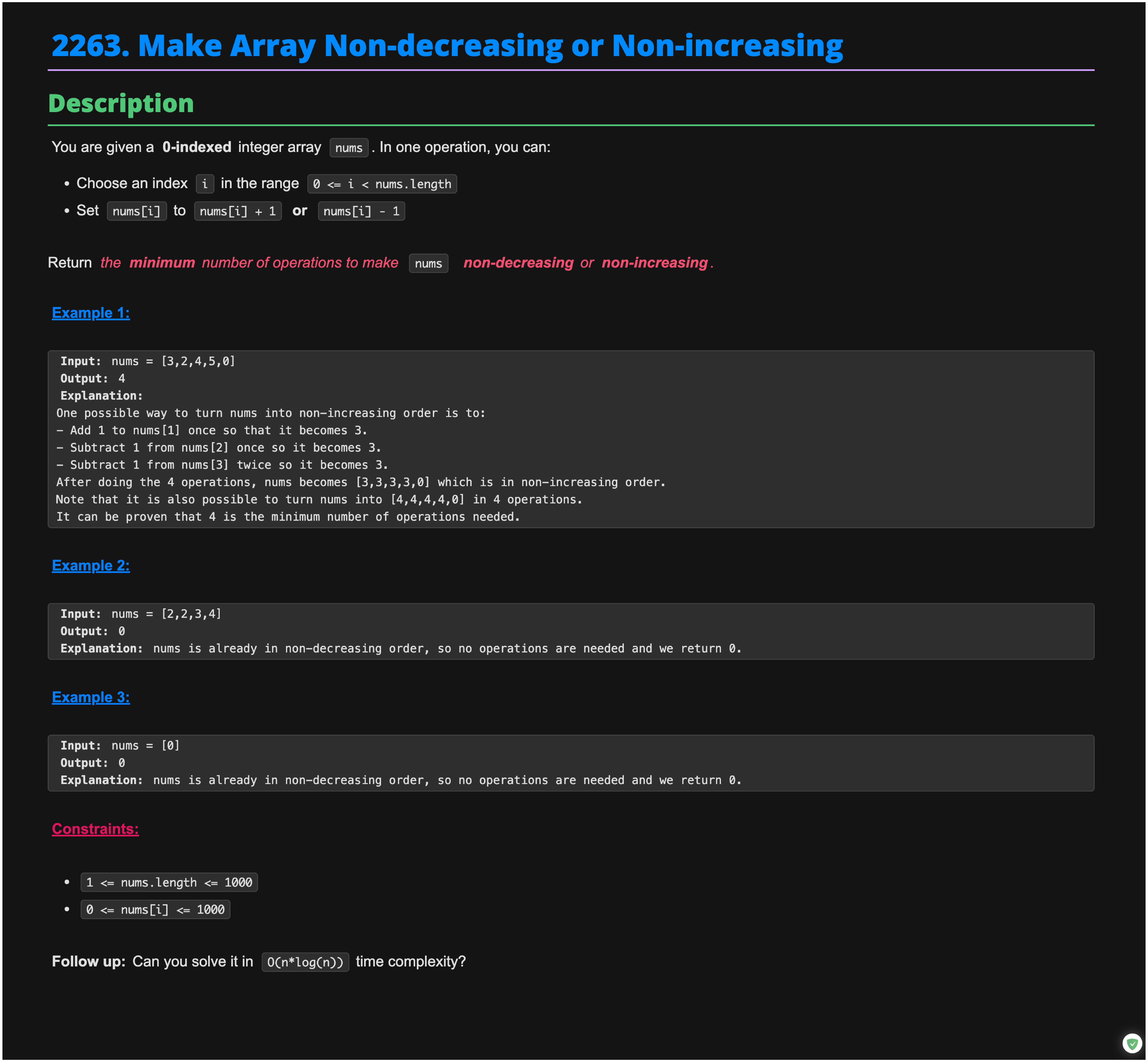

2263. Make Array Non-decreasing or Non-increasing

Description

You are given a **0-indexed** integer array `nums`. In one operation, you can:

- Choose an index `i` in the range `0 <= i < nums.length`
- Set `nums[i]` to `nums[i] + 1` **or** `nums[i] - 1`

Return *the* ***minimum*** *number of operations to make* `nums` ***non-decreasing*** *or* ***non-increasing***.

Example 1:

```
Input: nums = [3,2,4,5,0]
Output: 4
Explanation:
One possible way to turn nums into non-increasing order is to:
- Add 1 to nums[1] once so that it becomes 3.
- Subtract 1 from nums[2] once so it becomes 3.
- Subtract 1 from nums[3] twice so it becomes 3.
After doing the 4 operations, nums becomes [3,3,3,3,0] which is in non-increasing order.
Note that it is also possible to turn nums into [4,4,4,4,0] in 4 operations.
It can be proven that 4 is the minimum number of operations needed.
```

Example 2:

```
Input: nums = [2,2,3,4]
Output: 0
Explanation: nums is already in non-decreasing order, so no operations are needed and we return 0.
```

Example 3:

```
Input: nums = [0]
Output: 0
Explanation: nums is already in non-decreasing order, so no operations are needed and we return 0.
```

Constraints:

- `1 <= nums.length <= 1000`
- `0 <= nums[i] <= 1000`

Follow up: Can you solve it in `O(n*log(n))` time complexity?