

55. Jump Game

Medium

👍 10730

💬 615

♡ Add to List

🔗 Share

You are given an integer array `nums`. You are initially positioned at the array's **first index**, and each element in the array represents your maximum jump length at that position.

Return `true` if you can reach the last index, or `false` otherwise.

Example 1:

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

Output: `true`

Explanation: Jump 1 step from index 0 to 1, then 3 steps to the last index.

Example 2:

Input: `nums = [3,2,1,0,4]`

Output: `false`

Explanation: You will always arrive at index 3 no matter what. Its maximum jump length is 0, which makes it impossible to reach the last index.

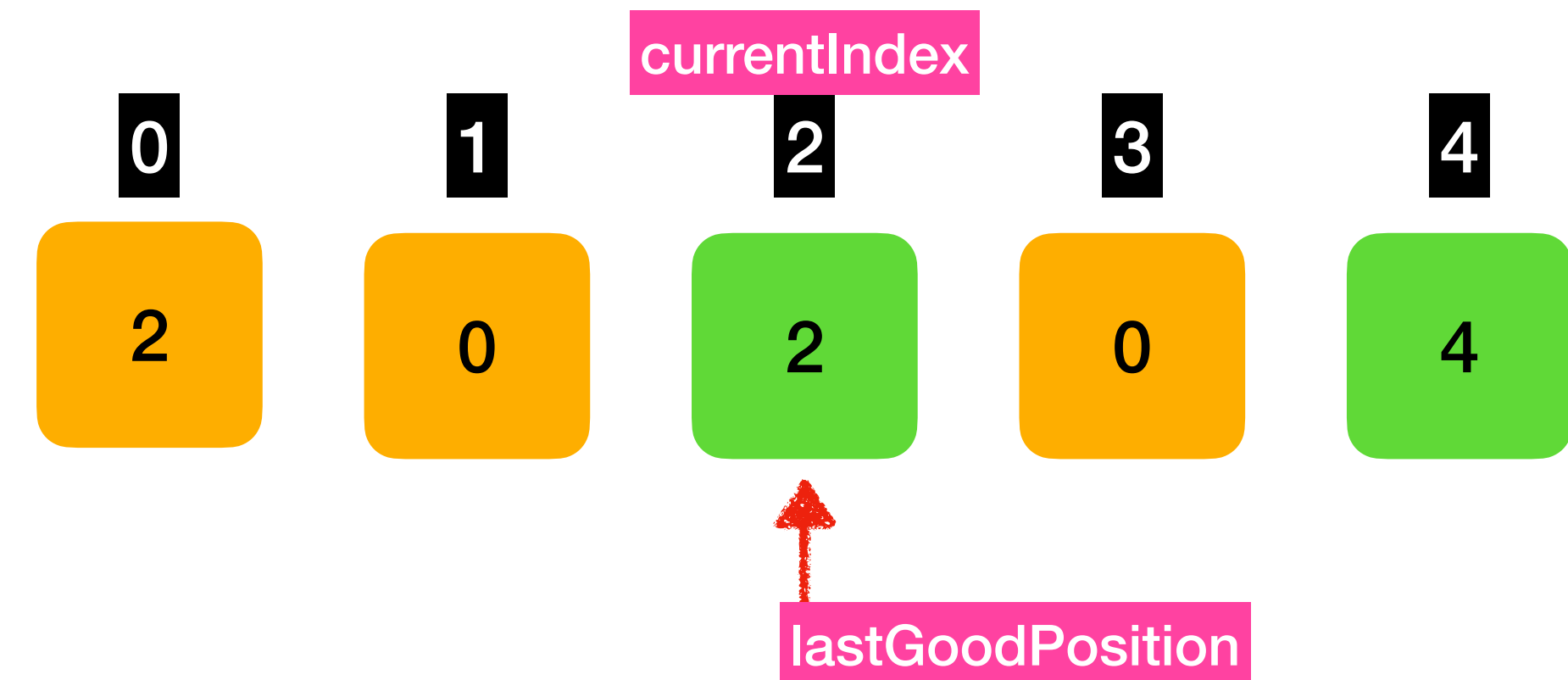
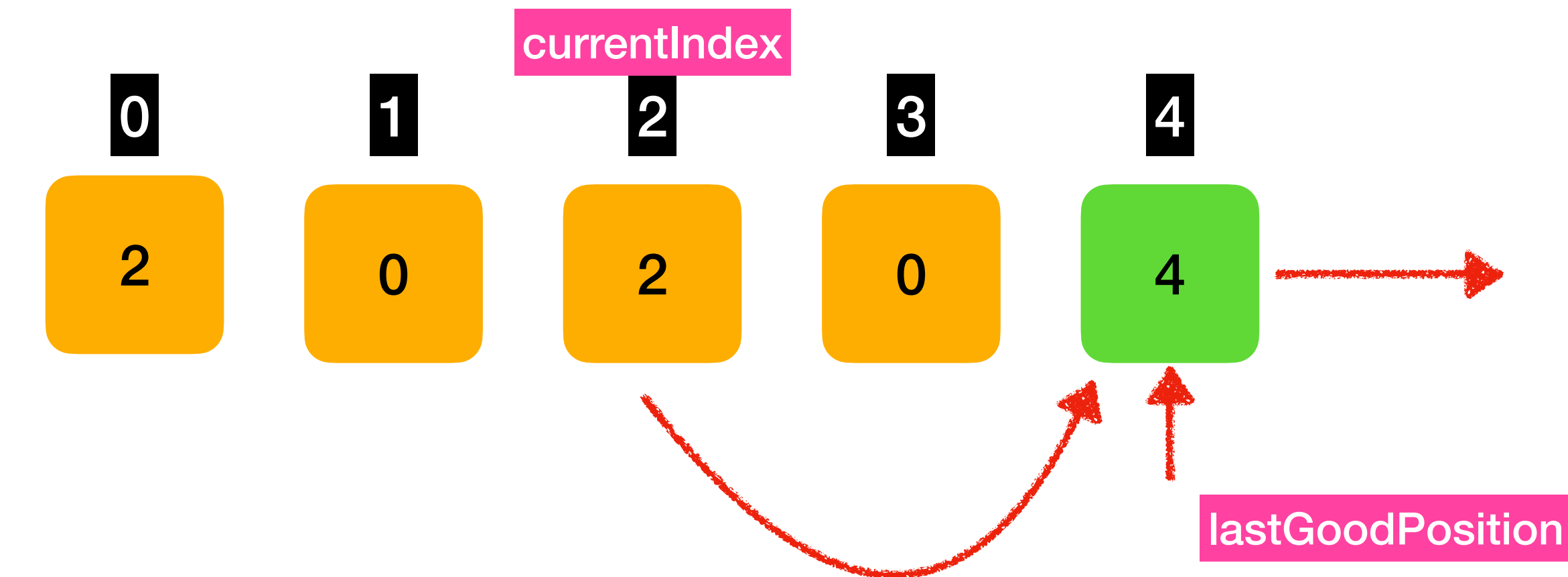
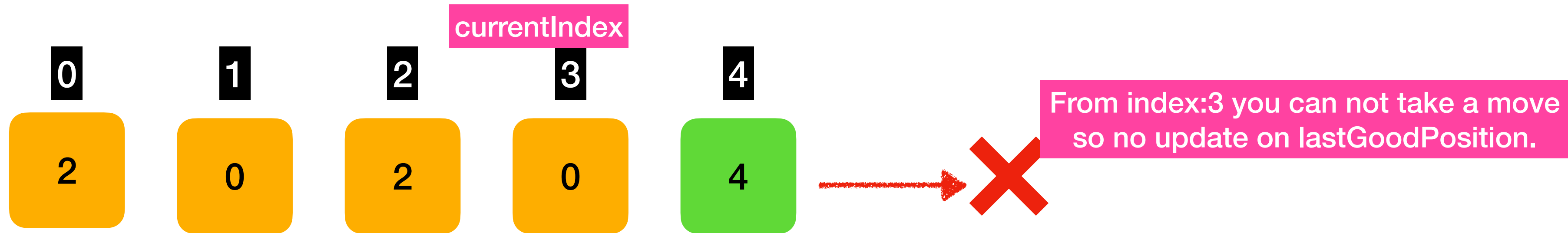
Constraints:

- `1 <= nums.length <= 104`
- `0 <= nums[i] <= 105`

Greedy Algorithm

Local Optimisation

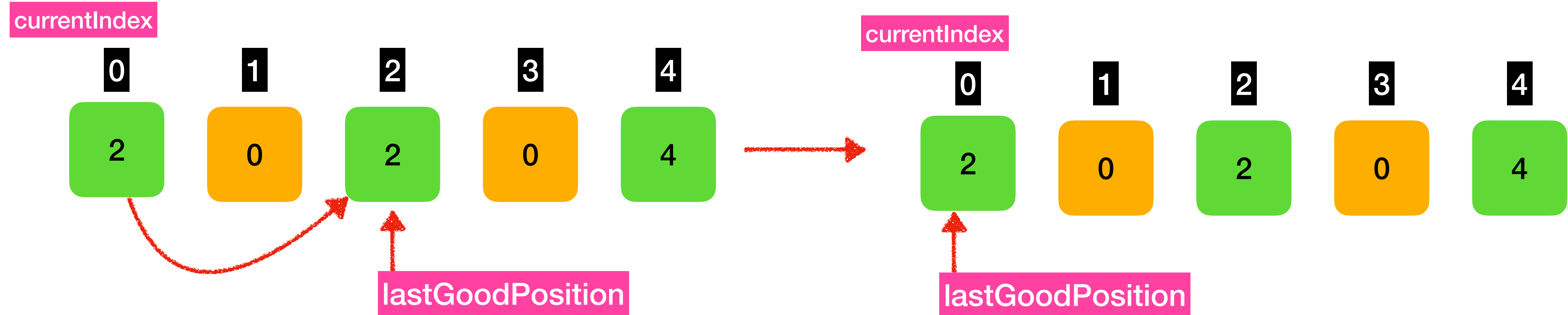
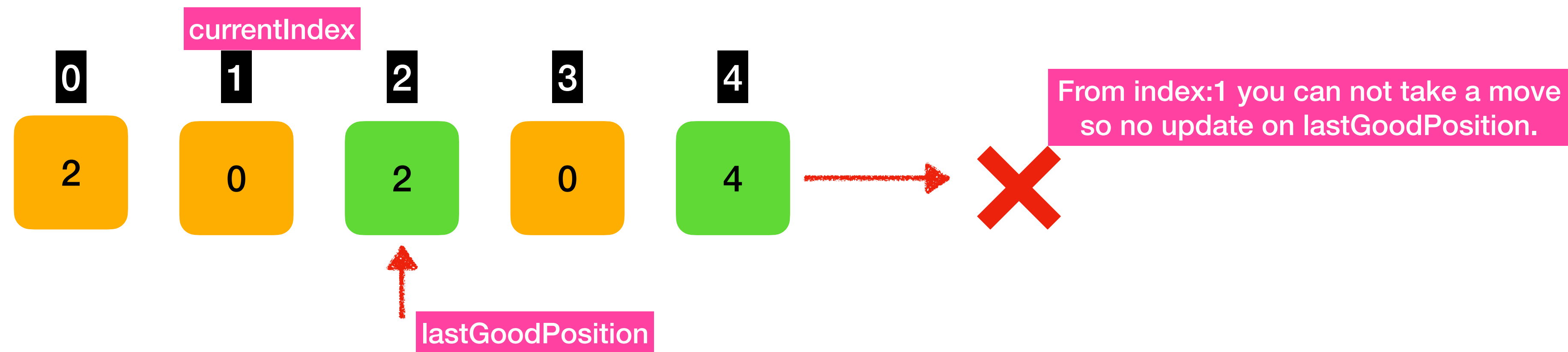
If you are $n-1$ index , you are in target. So always $(n-1)$ is the lastGoodIndex.



From index:2 with jump-2 you can reach to lastGoodPosition index.

So it means if we reach to index:2 we can reach to $n-1$

So do the local optimisation, make index:2 as lastGoodPosition.



From index:0 with jump-2 we can reach to lastGoodPosition index.

So it means if we reach to index:0 we can reach to n-1
So do the local optimisation, make index:0 as lastGoodPosition.

return lastGoodPosition == 0 ? True : False

Time Complexity : $O(n)$
Space Complexity : $O(1)$