Problem 3354: Make Array Elements Equal to Zero

Problem Information

Difficulty: Easy
Acceptance Rate: 68.35%
Paid Only: No
Tags: Array, Simulation, Prefix Sum

Problem Description

You are given an integer array `nums`.

Start by selecting a starting position `curr` such that `nums[curr] == 0`, and choose a movement **direction** of either left or right.

After that, you repeat the following process:

* If `curr` is out of the range `[0, n - 1]`, this process ends. * If `nums[curr] == 0`, move in the current direction by **incrementing** `curr` if you are moving right, or **decrementing** `curr` if you are moving left. * Else if `nums[curr] > 0`: * Decrement `nums[curr]` by 1. * **Reverse** your movement direction (left becomes right and vice versa). * Take a step in your new direction.

A selection of the initial position `curr` and movement direction is considered **valid** if every element in `nums` becomes 0 by the end of the process.

Return the number of possible **valid** selections.

Example 1:

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

Output: 2

Explanation:

The only possible valid selections are the following:

* Choose `curr = 3`, and a movement direction to the left. * `[1,0,2,**_0_** ,3] -> [1,0,**_2_** ,0,3] -> [1,0,1,**_0_** ,3] -> [1,0,1,0,**_3_**] -> [1,0,1,**_0_** ,2] -> [1,0,**_1_** ,0,2] -> [1,0,0,**_0_** ,2] -> [1,0,0,0,**_2_**] -> [1,0,0,**_0_** ,1] -> [1,0,**_0_** ,0,1] -> [1,**_0_** ,0,0,1] -> [**_1_** ,0,0,0,1] -> [0,**_0_** ,0,0,1] -> [0,0,**_0_** ,0,1] -> [0,0,0,**_0_** ,1] -> [0,0,0,0,**_1_**] -> [0,0,0,0,0]`. * Choose `curr = 3`, and a movement direction to the right. * `[1,0,2,**_0_** ,3] -> [1,0,2,0,**_3_**] -> [1,0,2,**_0_** ,2] -> [1,0,**_2_** ,0,2] -> [1,0,1,**_0_** ,2] -> [1,0,1,0,**_2_**] -> [1,0,1,**_0_** ,1] -> [1,0,**_1_** ,0,1] -> [1,0,0,**_0_** ,1] -> [1,0,0,0,**_1_**] -> [1,0,0,**_0_** ,0] -> [1,0,**_0_** ,0,0] -> [1,**_0_** ,0,0,0] -> [**_1_** ,0,0,0,0] -> [0,0,0,0,0].`

Example 2:

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

Output: 0

Explanation:

There are no possible valid selections.

Constraints:

* `1 <= nums.length <= 100` * `0 <= nums[i] <= 100` * There is at least one element `i` where `nums[i] == 0`.

Code Snippets

C++:

```
class Solution {
public:
int countValidSelections(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public int countValidSelections(int[] nums) {

}
}
```

Python3:

```
class Solution:
def countValidSelections(self, nums: List[int]) -> int:
```