Problem 2960: Count Tested Devices After Test Operations

Problem Information

Difficulty: Easy
Acceptance Rate: 78.68%
Paid Only: No
Tags: Array, Simulation, Counting

Problem Description

You are given a **0-indexed** integer array `batteryPercentages` having length `n`, denoting the battery percentages of `n` **0-indexed** devices.

Your task is to test each device `i` **in order** from `0` to `n - 1`, by performing the following test operations:

* If `batteryPercentages[i]` is **greater** than `0`: * **Increment** the count of tested devices. * **Decrease** the battery percentage of all devices with indices `j` in the range `[i + 1, n - 1]` by `1`, ensuring their battery percentage **never goes below** `0`, i.e, `batteryPercentages[j] = max(0, batteryPercentages[j] - 1)`. * Move to the next device. * Otherwise, move to the next device without performing any test.

Return _an integer denoting the number of devices that will be tested after performing the test operations in order._

Example 1:

Input: batteryPercentages = [1,1,2,1,3] **Output:** 3 **Explanation:** Performing the test operations in order starting from device 0: At device 0, batteryPercentages[0] > 0, so there is now 1 tested device, and batteryPercentages becomes [1,0,1,0,2]. At device 1, batteryPercentages[1] == 0, so we move to the next device without testing. At device 2, batteryPercentages[2] > 0, so there are now 2 tested devices, and batteryPercentages becomes [1,0,1,0,1]. At device 3, batteryPercentages[3] == 0, so we move to the next device without testing. At device 4, batteryPercentages[4] > 0, so there are now 3 tested devices, and batteryPercentages stays the same. So, the answer is 3.

Example 2:

Input: batteryPercentages = [0,1,2] **Output:** 2 **Explanation:** Performing the test operations in order starting from device 0: At device 0, batteryPercentages[0] == 0, so we move to the next device without testing. At device 1, batteryPercentages[1] > 0, so there is now 1 tested device, and batteryPercentages becomes [0,1,1]. At device 2, batteryPercentages[2] > 0, so there are now 2 tested devices, and batteryPercentages stays the same. So, the answer is 2.

Constraints:

* `1 <= n == batteryPercentages.length <= 100 ` * `0 <= batteryPercentages[i] <= 100`

Code Snippets

C++:

```
class Solution {
public:
int countTestedDevices(vector<int>& batteryPercentages) {

}
};
```

Java:

```
class Solution {
public int countTestedDevices(int[] batteryPercentages) {

}
}
```

Python3:

```
class Solution:
def countTestedDevices(self, batteryPercentages: List[int]) -> int:
```