Daily Coding Problem #79

Problem

This problem was asked by Facebook.

Given an array of integers, write a function to determine whether the array could become non-decreasing by modifying at most 1 element.

For example, given the array [10, 5, 7], you should return true, since we can modify the 10 into a 1 to make the array non-decreasing.

Given the array [10, 5, 1], you should return false, since we can't modify any one element to get a non-decreasing array.

Solution

In this problem, we can count each time an element goes down. Then, if it has went down more than twice, we can return

False right away. But if count is one, and the element is one that cannot be erased by modifying only one endpoint of that downtick, then we should return False as well.

```
def check(lst):
    count = 0
    for i in range(len(lst) - 1):
        if lst[i] > lst[i + 1]:
            if count > 0:
                return False
            if i - 1 >= 0 and i + 2 < len(lst) and lst[i] > lst[i + 2] and lst[i + 1] < lst[i - 1]:
                return False
            count += 1
    return True
```