581. Shortest Unsorted Continuous Subarray

Description | Hints | Submissions | Discuss | Solution

Pick One

Given an integer array, you need to find one **continuous subarray** that if you only sort this subarray in ascending order, then the whole array will be sorted in ascending order, too.

You need to find the **shortest** such subarray and output its length.

Example 1:

```
Input: [2, 6, 4, 8, 10, 9, 15]
Output: 5
Explanation: You need to sort [6, 4, 8, 10, 9] in ascending order to make the whole array sorted in ascending order.
```

Note:

1. Then length of the input array is in range [1, 10,000].
2. The input array may contain duplicates, so ascending order here means <=.

```java
public class L581 {
    /*
     * 先用一个数组tmp保存nums，然后对tmp进行排序，
     * 然后用两个变量low和high去找两个数组出现不同处的
     * 第一个位置和最后一个位置，最后返回high-low+1就是要找的
     * 数组长度。
     */
     public int findUnsortedSubarray(int[] nums) {
        //深拷贝
         int [] tmp = nums.clone();
         Arrays.sort(tmp);

         int n = nums.length;
         int low = 0;
         int high = n - 1;

         while (low < n && nums[low] == tmp[low]) {
            low ++;
        }
         while (high >= low && nums[high] == tmp[high]) {
            high --;
        }
         return high - low + 1;
    }
}
```