

给定一个整数数组，你需要寻找一个**连续的子数组**，如果对这个子数组进行升序排序，那么整个数组都会变为升序排序。

你找到的子数组应是**最短的**，请输出它的长度。

示例 1:

输入: [2, 6, 4, 8, 10, 9, 15]

输出: 5

解释: 你只需要对 [6, 4, 8, 10, 9] 进行升序排序，那么整个表都会变为升序排序。

说明:

1. 输入的数组长度范围在 [1, 10,000]。
2. 输入的数组可能包含**重复元素**，所以**升序**的意思是 \leq 。

思路，从左右开始遍历数组，记录左边的最大值和右边的最小值，若是左边的小于最大值，那么他就不该存在该位置上，end标记，若右边大于最小值，同理star标记。返回end-star+1;

```
class Solution {
public:
    int findUnsortedSubarray(vector<int>& nums) {
        int n=nums.size();
        int min=nums[n-1];
        int max=nums[0];
        int end=-2;
        int star=-1;
        for(int i=0;i<n;++i){
            min=min<nums[n-1-i]?min:nums[n-1-i];
            max=max>nums[i]?max:nums[i];
            if(min<nums[n-1-i]){
                star=n-1-i;
            }
            if(max>nums[i]){
                end=i;
            }
        }
        return end-star+1;
    }
};
```

}

};