

最短无序连续子数组

给你一个整数数组 `nums`，你需要找出一个 **连续子数组**，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。

请你找出符合题意的 **最短** 子数组，并输出它的长度。

示例 1:

输入: `nums = [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。

示例 2:

输入: `nums = [1,2,3,4]`

输出: 0

示例 3:

输入: `nums = [1]`

输出: 0

```
int cmp(const void* a,const void* b)
{
    return *(int*)a-*(int*)b;
}

int findUnsortedSubarray(int* nums, int numsSize){
    //拷贝一个数组
    int* arr=(int*)malloc(sizeof(int)*numsSize);
    memcpy(arr,nums,sizeof(int)*numsSize);
    //新数组排序(升序)
    qsort(arr,numsSize,sizeof(int),cmp);
    //从前向后遍历记录排序数组与原数组第一个不同位置
    int start=0;
    int end=0;
    for(int i=0;i<numsSize;i++)
    {
        if(arr[i]!=nums[i])
        {
            start=i;
            end=i;
        }
    }
    return end-start+1;
}
```

```
        break;
    }
}
//从前向后遍历记录排序数组与原数组第一个不同位置
for(int j=numsSize-1;j>=0;j--)
{
    if(arr[j]!=nums[j])
    {
        end=j;
        break;
    }
}
if(start==end)
    return 0;
else
    return end-start+1;//计算起始位置距离
}
```