

0795. 区间子数组个数

👤 ITCharge ⌚ 大约 1 分钟

- 标签：数组、双指针
- 难度：中等

题目链接

- [0795. 区间子数组个数 - 力扣](#)

题目大意

给定一个元素都是正整数的数组 A ，正整数 L 以及 R ($L \leq R$)。

求连续、非空且其中最大元素满足大于等于 L 小于等于 R 的子数组个数。

解题思路

最大元素满足大于等于 L 小于等于 R 的子数组个数 = 最大元素小于等于 R 的子数组个数 - 最大元素小于 L 的子数组个数。

其中「最大元素小于 L 的子数组个数」也可以转变为「最大元素小于等于 $L - 1$ 的子数组个数」。那么现在的问题就变为了如何计算最大元素小于等于 k 的子数组个数。

我们使用 $count$ 记录小于等于 k 的连续元素数量，遍历一遍数组，如果遇到 $nums[i] \leq k$ 时， $count$ 累加，表示在此位置上结束的有效子数组数量为 $count + 1$ 。如果遇到 $nums[i] > k$ 时， $count$ 重新开始计算。每次遍历完将有效子数组数量累加到答案中。

代码

```
class Solution:
    def numSubarrayMaxK(self, nums, k):
        ans = 0
        count = 0
        for i in range(len(nums)):
```

py

```
        if nums[i] <= k:
            count += 1
        else:
            count = 0
        ans += count
    return ans

    def numSubarrayBoundedMax(self, nums: List[int], left: int, right: int) ->
int:
        return self.numSubarrayMaxK(nums, right) - self.numSubarrayMaxK(nums,
left - 1)
```