0209. 长度最小的子数组

ITCharge 大约 2 分钟

- 标签：数组、二分查找、前缀和、滑动窗口
- 难度：中等

题目链接

- 0209. 长度最小的子数组 - 力扣

题目大意

描述：给定一个只包含正整数的数组 $nums$ 和一个正整数 $target$。

要求：找出数组中满足和大于等于 $target$ 的长度最小的「连续子数组」，并返回其长度。如果不存在符合条件的子数组，返回 0。

说明：

- $1 \leq target \leq 10^9$。
- $1 \leq nums.length \leq 10^5$。
- $1 \leq nums[i] \leq 10^5$。

示例：

- 示例 1：

```py
输入：target = 7, nums = [2,3,1,2,4,3]
输出：2
解释：子数组 [4,3] 是该条件下的长度最小的子数组。
```

- 示例 2：

```py
输入：target = 4, nums = [1,4,4]
输出：1
```

解题思路

思路 1：滑动窗口（不定长度）

最直接的做法是暴力枚举，时间复杂度为 $O(n^2)$。但是我们可以利用滑动窗口的方法，在时间复杂度为 $O(n)$ 的范围内解决问题。

用滑动窗口来记录连续子数组的和，设定两个指针：$left$、$right$，分别指向滑动窗口的左右边界，保证窗口中的和刚好大于等于 $target$。

1. 一开始，$left$、$right$ 都指向 0。
2. 向右移动 $right$，将最右侧元素加入当前窗口和 $window\_sum$ 中。
3. 如果 $window\_sum \geq target$，则不断右移 $left$，缩小滑动窗口长度，并更新窗口和的最小值，直到 $window\_sum < target$。
4. 然后继续右移 $right$，直到 $right \geq len(nums)$ 结束。
5. 输出窗口和的最小值作为答案。

思路 1：代码

py

```py
class Solution:
    def minSubArrayLen(self, target: int, nums: List[int]) -> int:
        size = len(nums)
        ans = size + 1
        left = 0
        right = 0
        window_sum = 0

        while right < size:
            window_sum += nums[right]

            while window_sum >= target:
                ans = min(ans, right - left + 1)
                window_sum -= nums[left]
```

```
                left += 1

            right += 1

        return ans if ans != size + 1 else 0
```

思路 1：复杂度分析

- **时间复杂度**：$O(n)$。
- **空间复杂度**：$O(1)$。