

209 长度最小的子数组

问题描述:

给定一个含有 n 个正整数的数组和一个正整数 $target$ 。

找出该数组中满足其和 $\geq target$ 的长度最小的连续子数组 $[nums_l, nums_l+1, \dots, nums_r-1, nums_r]$ ，并返回其长度。如果不存在符合条件的子数组，返回 0。

解法：滑动窗口

```
class Solution {
public:
    int minSubArrayLen(int target, vector<int>& nums) {
        int result = INT32_MAX;
        int i = 0; // 窗口起始位置
        int len = 0; // 窗口长度
        int sums = 0; // 窗口总和

        for (int j = 0; j < nums.size(); j++) {
            sums += nums[j]; // 配合i++实现窗口的滑动，避免不必要的操作
            while (sums >= target) {
                len = j - i + 1;
                result = result > len ? len : result;
                sums -= nums[i++];
            }
        }
        return result == INT32_MAX ? 0 : result; // 如果result没有被赋值过，说明没有符合
        条件的子序列
    }
};
```

时间复杂度: $O(n)$

//不要以为for里放一个while就以为是 $O(n^2)$ 啊，主要是看每一个元素被操作的次数，每个元素在滑动窗后进来操作一次，出去操作一次，每个元素都是被操作两次，所以时间复杂度是 $2 \times n$ 也就是 $O(n)$ 。

空间复杂度: $o(1)$

相关题目 904 76

904 水果成篮

