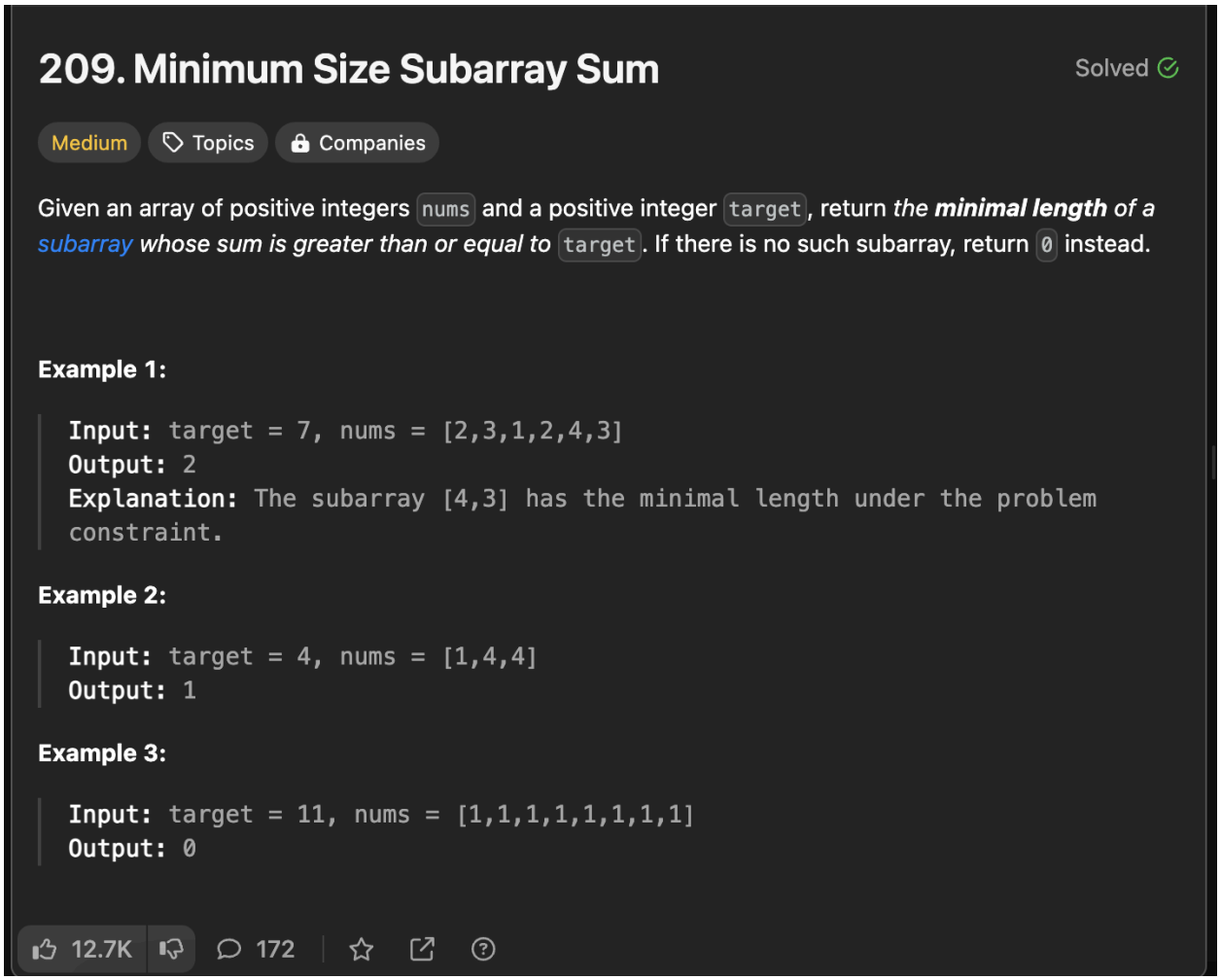
6: Leetcode 209 Minimum Size Subarray Sum

⇒https://leetcode.com/problems/minimum-size-subarray-sum/description/?envType=list&envId=xlep8di5

Description



209. Minimum Size Subarray Sum

Solved

Medium | Topics | Companies

Given an array of positive integers `nums` and a positive integer `target`, return *the* ***minimal length*** *of a subarray whose sum is greater than or equal to* `target`. If there is no such subarray, return `0` instead.

Example 1:

> **Input:** target = 7, nums = [2,3,1,2,4,3]
> **Output:** 2
> **Explanation:** The subarray [4,3] has the minimal length under the problem constraint.

Example 2:

> **Input:** target = 4, nums = [1,4,4]
> **Output:** 1

Example 3:

> **Input:** target = 11, nums = [1,1,1,1,1,1,1,1]
> **Output:** 0

12.7K | 172

2 - Code

```
from sys import maxsize

class Solution:
    def minSubArrayLen(self, target: int, nums: List[int]) -> int:
        left,currentSum = 0,0
```

```
        res = maxsize

        for i in range(0,len(nums)):
            currentSum += nums[i]
            while currentSum >= target:
                res = min(res, i - left +1)
                currentSum -= nums[left]
                left += 1
        return res if res!= maxsize else 0
```

3 - Notes-

64 209: Minimum size subarray sum:

→ sliding window:

[2, 3, 1, 2, 4, 3]
0 1 2 3 4 5

left, right=1, r-

res = max 0 0 SUM 2

= max l=0, r=1, SUM → 5 (less than target 7), add right, r++

= max l=0, r=2 " → 6 " , r++

= 3-0+1 = 4 l=0, r=3 " → 8 (more than 7) subtract remove left, l++

= 4 l=1, r=3 " → 6 (less) r++

= 4-1+1 = 4 l=1, r=4 " → 10 (more)

= 4-2+1 = 3 2, r=4 → 7 → if matched store r-2+1, do l++

= 5-3+1 = 3 3, r=4 → 6 → less, r++

3, r=5 → 8 → more, l++

= 5-4+1 = 2 4, r=5 → 7 → store (r-l+1), keep min.

→ min.

Algo: 1) res = max, sum = 0, left = 0.

2) for i in 0 to len(nums) →

right side ~~initial sum~~ → sum = sum + nums[i]

inc sum

while sum >= 5:

res (update) → min(res, i - l + 1)

sum = sum - nums[left]

left ++