2302. Count Subarrays With Score Less Than K

Attempted

Hard | Topics | Companies | Hint

The **score** of an array is defined as the **product** of its sum and its length.

- For example, the score of `[1, 2, 3, 4, 5]` is `(1 + 2 + 3 + 4 + 5) * 5 = 75`.

Given a positive integer array `nums` and an integer `k`, return *the **number of non-empty subarrays** of `nums` whose score is **strictly less** than `k`*.

A **subarray** is a contiguous sequence of elements within an array.

Example 1:

```
Input: nums = [2,1,4,3,5], k = 10
Output: 6
Explanation:
The 6 subarrays having scores less than 10 are:
- [2] with score 2 * 1 = 2.
- [1] with score 1 * 1 = 1.
- [4] with score 4 * 1 = 4.
- [3] with score 3 * 1 = 3.
- [5] with score 5 * 1 = 5.
- [2,1] with score (2 + 1) * 2 = 6.
Note that subarrays such as [1,4] and [4,3,5] are not considered
because their scores are 10 and 36 respectively, while we need scores
strictly less than 10.
```

Python3 • Auto

```python
class Solution:
    def countSubarrays(self, nums: List[int], k: int) -> int:
        res = 0
        for i in range(len(nums)):
            l = 0
            s = 0
            for j in range(i, len(nums)):
                l += 1
                s += nums[j]
                if l * s >= k:
                    break
                else:
                    res += 1

        return res
```

Ln 9, Col 28 | Saved

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

Time Limit Exceeded

sliding window works

Accepted 167 / 167 testcases passed

AndrewC275 submitted at Apr 29, 2025 15:16

Editorial | Solution

Runtime

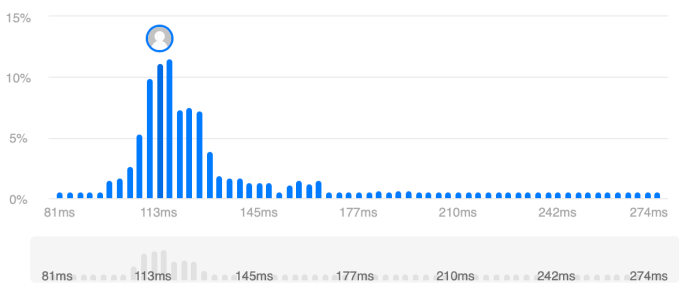
115 ms | Beats 74.54%

Analyze Complexity

Memory

30.40 MB | Beats 87.42%

Analyze Complexity




```python
class Solution:
    def countSubarrays(self, nums: List[int], k: int) -> int:
        n = len(nums)
        res, s = 0, 0
        l = 0

        for r in range(n):
            s += nums[r]
            while l <= r and s * (r - l + 1) >= k:
                s -= nums[l]
                l += 1
            res += r - l + 1

        return res
```

Ln 12, Col 25 | Saved

for ...

while ... (mostly use the exception to terminate the increasing window size)