

325. Maximum Size Subarray Sum Equals k Premium

Medium

Topics

Companies

Hint

Given an integer array `nums` and an integer `k`, return the *maximum length of a subarray that sums to k*. If there is not one, return `0` instead.

Example 1:

Input: `nums = [1,-1,5,-2,3]`, `k = 3`

Output: `4`

Explanation: The subarray `[1, -1, 5, -2]` sums to 3 and is the longest.

- Sliding window ✗
- commutative sum. ✓

Accepted 37 / 37 testcases passed

AndrewC275 submitted at Apr 17, 2025 09:36

Editorial Solution

Runtime

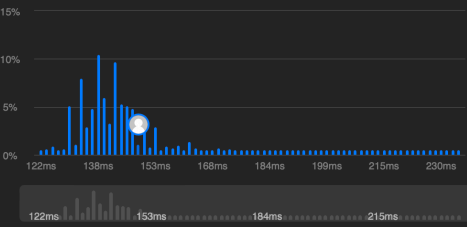
148 ms | Beats 34.29%

Analyze Complexity

Memory

63.04 MB | Beats 78.05%

Analyze Complexity



Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

```
1 class Solution:
2     def maxSubArrayLen(self, nums: List[int], k: int) -> int:
3         curr = 0
4         res = 0
5         seen = {}
6
7         for i in range(len(nums)):
8             curr += nums[i]
9
10            if curr == k:
11                res = i + 1
12
13            if curr - k in seen:
14                res = max(res, i - seen[curr - k])
15
16            if curr not in seen:
17                seen[curr] = i
18
```

Ln 14, Col 36 Saved

no need to
maintain an
array to
store sums.