

Problem List

DescriptionEditorialSolutionsSubmissions

346. Moving Average from Data StreamPremium

Solved

EasyTopicsCompanies

Given a stream of integers and a window size, calculate the moving average of all integers in the sliding window.

Implement the `MovingAverage` class:

MovingAverage(int size) Initializes the object with the size of the window size.

double next(int val) Returns the moving average of the last size values of the stream.

Example 1:

Input

["MovingAverage", "next", "next", "next", "next"]
[[3], [1], [10], [3], [5]]

Output

[null, 1.0, 5.5, 4.66667, 6.0]

Explanation

MovingAverage movingAverage = new MovingAverage(3);
movingAverage.next(1); // return 1.0 = 1 / 1
movingAverage.next(10); // return 5.5 = (1 + 10) / 2
movingAverage.next(3); // return 4.66667 = (1 + 10 + 3) / 3
movingAverage.next(5); // return 6.0 = (10 + 3 + 5) / 3

Constraints:

1 <= size <= 1000

-10⁵ <= val <= 10⁵

At most 10⁴ calls will be made to next.

Seen this question in a real interview before? 1/5

YesNo

Accepted 377.5K | Submissions 482.1K | Acceptance Rate 78.3%

TopicsCompaniesSimilar QuestionsDiscussion (12)

Copyright © 2024 LeetCode All rights reserved

1.6K12

</> Code

Python3Auto

```
1 class MovingAverage:
2
3     def __init__(self, size: int):
4         self.a = []
5         self.size = size
6
7
8     def next(self, val: int) -> float:
9         self.a.append(val)
10
11         return sum(self.a[-self.size:])/min(len(self.a), self.size)
12
13
14
15 # Your MovingAverage object will be instantiated and called as such:
16 # obj = MovingAverage(size)
17 # param_1 = obj.next(val)
```

SavedLn 6, Col 9

TestcaseTest Result

AcceptedRuntime: 41 ms

Case 1

Input

["MovingAverage", "next", "next", "next", "next"]

[[3], [1], [10], [3], [5]]

Output

[null, 1.00000, 5.50000, 4.66667, 6.00000]

Expected

[null, 1.00000, 5.50000, 4.66667, 6.00000]