

Problem List

RunSubmit

DescriptionAcceptedEditorialSubmissionsSolutions

251. Flatten 2D Vector

Premium

Solved

Medium

TopicsCompaniesHint

Design an iterator to flatten a 2D vector. It should support the `next` and `hasNext` operations.

Implement the `Vector2D` class:

•

`Vector2D(int[][] vec)` initializes the object with the 2D vector `vec`.

•

`next()` returns the next element from the 2D vector and moves the pointer one step forward. You may assume that all the calls to `next` are valid.

•

`hasNext()` returns `true` if there are still some elements in the vector, and `false` otherwise.

Example 1:

Input

["Vector2D", "next", "next", "next", "hasNext", "hasNext", "next", "hasNext"]
[[[1, 2], [3], [4]], [], [], [], [], [], []]

Output

[null, 1, 2, 3, true, true, 4, false]

Explanation

Vector2D vector2D = new Vector2D([[1, 2], [3], [4]]);
vector2D.next(); // return 1
vector2D.next(); // return 2
vector2D.next(); // return 3
vector2D.hasNext(); // return True
vector2D.hasNext(); // return True
vector2D.next(); // return 4
vector2D.hasNext(); // return False

Constraints:

• 0 <= vec.length <= 200

• 0 <= vec[i].length <= 500

• -500 <= vec[i][j] <= 500

• At most 10⁵ calls will be made to `next` and `hasNext`.

Follow up:

As an added challenge, try to code it using only [iterators in C++](#) or [iterators in Java](#).

Seen this question in a real interview before?

1/5

Yes

No

Accepted

126.5K

Submissions

255.5K

Acceptance Rate

49.5%

Topics

Companies

Hint 1

Hint 2

Hint 3

Hint 4

Hint 5

Hint 6

Hint 7

Similar Questions

Discussion (6)

Copyright © 2024 LeetCode All rights reserved

711

6

6

</> Code

Python3

```
1 class Vector2D:
2
3     def __init__(self, vec: List[List[int]]):
4         self.listx = []
5
6         for each in vec:
7             self.listx.extend(each)
8
9         self.listx = self.listx[::-1]
10
11
12     def next(self) -> int:
13         return self.listx.pop()
14
15
16     def hasNext(self) -> bool:
17         if self.listx:
18             return True
19         else:
20             return False
21
22
23
24 # Your Vector2D object will be instantiated and called as such:
25 # obj = Vector2D(vec)
26 # param_1 = obj.next()
27 # param_2 = obj.hasNext()
```

Saved

Ln 11, Col 1

Testcase

Test Result

Accepted

Runtime: 37 ms

Case 1

Input

["Vector2D", "next", "next", "next", "hasNext", "hasNext", "next", "hasNext"]

[[[1,2],[3],[4]],[],[],[],[],[],[]]