251. Flatten 2D Vector Premium

Medium | Topics | Companies | Hint

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 10^5 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 131.5K | Submissions 264.4K | Acceptance Rate 49.7%

Topics

Array Two Pointers Design Iterator

Companies

0 - 3 months

Airbnb 3

0 - 6 months

Google 3 X 2 Zenefits 2

6 months ago

Amazon 2

Hint 1

How many variables do you need to keep track?

Hint 2

Two variables is all you need. Try with `x` and `y`.

Hint 3

Beware of empty rows. It could be the first few rows.

Hint 4

To write correct code, think about the invariant to maintain. What is it?

Hint 5

The invariant is `x` and `y` must always point to a valid point in the 2d vector. Should you maintain your invariant *ahead of time* or *right when you need it*?

Hint 6

Not sure? Think about how you would implement `hasNext()`. Which is more complex?

Hint 7

Common logic in two different places should be refactored into a common method.

Similar Questions

Binary Search Tree Iterator	Medium
Zigzag Iterator	Medium
Peeking Iterator	Medium
Flatten Nested List Iterator	Medium

Discussion (7)