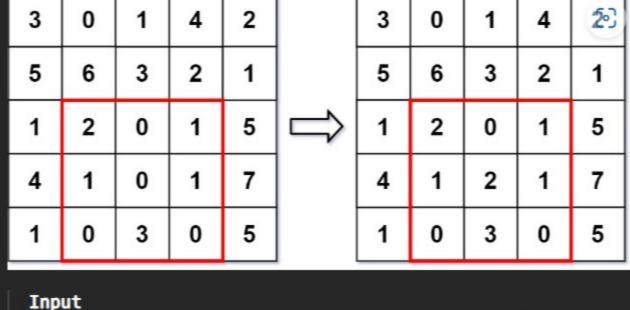
308. Range Sum Query 2D - Mutable Premium **E** Companies ♥ Topics Given a 2D matrix matrix, handle multiple queries of the following types: Update the value of a cell in matrix. Calculate the sum of the elements of matrix inside the rectangle defined by its upper left corner (row1, col1) and lower right corner (row2, col2). Implement the NumMatrix class: NumMatrix(int[][] matrix) Initializes the object with the integer matrix matrix.

- void update(int row, int col, int val) Updates the value of matrix[row][col] to be val.
- int sumRegion(int row1, int col1, int row2, int col2) Returns the sum of the elements of matrix inside the rectangle defined by its upper left corner (row1, col1) and lower right corner (row2, col2).

Example 1:



["NumMatrix", "sumRegion", "update", "sumRegion"]

```
Output
[null, 8, null, 10]
Explanation
NumMatrix numMatrix = new NumMatrix([[3, 0, 1, 4, 2], [5, 6, 3, 2,
```

[[[[3, 0, 1, 4, 2], [5, 6, 3, 2, 1], [1, 2, 0, 1, 5], [4, 1, 0, 1,

7], [1, 0, 3, 0, 5]]], [2, 1, 4, 3], [3, 2, 2], [2, 1, 4, 3]]

1], [1, 2, 0, 1, 5], [4, 1, 0, 1, 7], [1, 0, 3, 0, 5]]);

numMatrix.sumRegion(2, 1, 4, 3); // return 8 (i.e. sum of the left red rectangle) numMatrix.update(3, 2, 2); // matrix changes from left image to right image numMatrix.sumRegion(2, 1, 4, 3); // return 10 (i.e. sum of the right red rectangle)

m == matrix.length

Constraints:

• 1 <= m, n <= 200

n == matrix[i].length

- -1000 <= matrix[i][j] <= 1000
- 0 <= col < n

0 <= row < m

0 <= row1 <= row2 < m

-1000 <= val <= 1000

At most 5000 calls will be made to sumRegion and update.

0 <= col1 <= col2 < n

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

Accepted 78.9K Submissions 177.6K Acceptance Rate 44.4%

No

Yes

Topics Array Design Binary Indexed Tree Segment Tree Matrix

Companies 0 - 3 months Google 2

Bloomberg 7

6 months ago

₩ Similar Questions

Discussion (5)

Range Sum Query - Mutable

Sum of Matrix After Queries

Range Sum Query 2D - Immutable

Increment Submatrices by One

Medium

Medium

Medium

Medium

Copyright © 2024 LeetCode All rights reserved