

## 2536. Increment Submatrices by One

[My Submissions \(/contest/weekly-contest-328/problems/increment-submatrices-by-one/submissions/\)](#)
[Back to Contest \(/contest/weekly-contest-328/\)](#)

You are given a positive integer  $n$ , indicating that we initially have an  $n \times n$  **0-indexed** integer matrix `mat` filled with zeroes.

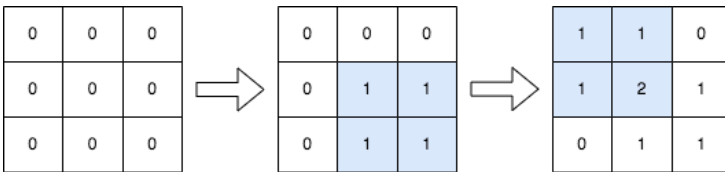
You are also given a 2D integer array `query`. For each `query[i] = [row1i, col1i, row2i, col2i]`, you should do the following operation:

- Add 1 to **every element** in the submatrix with the **top left** corner  $(row1_i, col1_i)$  and the **bottom right** corner  $(row2_i, col2_i)$ . That is, add 1 to `mat[x][y]` for all  $row1_i \leq x \leq row2_i$  and  $col1_i \leq y \leq col2_i$ .

Return the matrix `mat` after performing every query.

User Accepted:	5940
User Tried:	9208
Total Accepted:	6134
Total Submissions:	16457
Difficulty:	Medium

## Example 1:



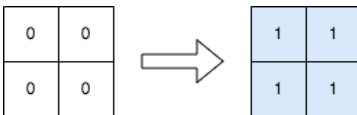
**Input:**  $n = 3$ , `queries = [[1,1,2,2],[0,0,1,1]]`

**Output:** `[[1,1,0],[1,2,1],[0,1,1]]`

**Explanation:** The diagram above shows the initial matrix, the matrix after the first query, and the matrix after the second query.

- In the first query, we add 1 to every element in the submatrix with the top left corner  $(1, 1)$  and bottom right corner  $(2, 2)$ .
- In the second query, we add 1 to every element in the submatrix with the top left corner  $(0, 0)$  and bottom right corner  $(1, 1)$ .

## Example 2:



**Input:**  $n = 2$ , `queries = [[0,0,1,1]]`

**Output:** `[[1,1],[1,1]]`

**Explanation:** The diagram above shows the initial matrix and the matrix after the first query.

- In the first query we add 1 to every element in the matrix.

## Constraints:

- $1 \leq n \leq 500$
- $1 \leq \text{queries.length} \leq 10^4$
- $0 \leq row1_i \leq row2_i < n$
- $0 \leq col1_i \leq col2_i < n$

[Discuss \(https://leetcode.com/problems/increment-submatrices-by-one/discuss/\)](https://leetcode.com/problems/increment-submatrices-by-one/discuss/)

JavaScript



```

1 function DiffArray2D(g) {
2   let n = g.length, m = g[0].length, d = [...Array(n)].map(() => Array(m + 1).fill(0));
3   initialize();
4   return { addRange, recover, D }
5   function initialize() {
6     for (let i = 0; i < n; i++) d[i][0] = g[i][0];
7     for (let i = 0; i < n; i++)
8       for (let j = 1; j < m; j++)
9         d[i][j] = g[i][j] - g[i][j - 1];
10  }

```

```
11 function addRange(x1, y1, x2, y2, v) {
12     for (let i = x1; i <= x2; i++) {
13         d[i][y1] += v;
14         d[i][y2 + 1] -= v;
15     }
16 }
17 function recover() {
18     let res = [...Array(n)].map(() => Array(m).fill(0));
19     for (let i = 0; i < n; i++)
20         for (let j = 0; j < m; j++)
21             res[i][j] = j == 0 ? d[i][j] : d[i][j] + res[i][j - 1];
22     return res;
23 }
24 function D() {
25     return d;
26 }
27 }
28
29 const rangeAddQueries = (n, queries) => {
30     let g = [...Array(n)].map(() => Array(n).fill(0)), da = new DiffArray2D(g);
31     for (const [x1, y1, x2, y2] of queries) da.addRange(x1, y1, x2, y2, 1);
32     let res = da.recover();
33     return res;
34 };
```

☐ Custom TestcaseSubmission Result: **Accepted** (/submissions/detail/1028011637/) ?[More Details >](/submissions/detail/1028011637/)

Share your acceptance!

Copyright © 2023 LeetCode

[Help Center \(/support/\)](/support/) | [Jobs \(/jobs\)](/jobs/) | [Bug Bounty \(/bugbounty\)](/bugbounty/) | [Online Interview \(/interview/\)](/interview/) | [Students \(/student\)](/student/) | [Terms \(/terms\)](/terms/) | [Privacy Policy \(/privacy\)](/privacy/)[United States \(/region\)](/region/)