6472. Sum of Matrix After Queries

My Submissions (/contest/weekly-contest-348/problems/sum-of-matrix-after-queries/submissions/) Back to Contest (/contest/weekly-contest-348/)

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

You are given an integer `n` and a **0-indexed 2D array** `queries` where `queries[i] = [type_i, index_i, val_i]`.

Initially, there is a **0-indexed** `n x n` matrix filled with `0`'s. For each query, you must apply one of the following changes:

- if `type_i == 0`, set the values in the row with `index_i` to `val_i`, overwriting any previous values.
- if `type_i == 1`, set the values in the column with `index_i` to `val_i`, overwriting any previous values.

Return *the sum of integers in the matrix after all queries are applied*.

Example 1:

Initial Matrix

0	0	0
0	0	0
0	0	0

Query 0

1	1	1
0	0	0
0	0	0

Query 1

1	1	2
0	0	2
0	0	2

Query 2

1	1	2
0	0	2
3	3	3

Query 3

4	1	2
4	0	2
4	3	3

```
Input: n = 3, queries = [[0,0,1],[1,2,2],[0,2,3],[1,0,4]]
Output: 23
Explanation: The image above describes the matrix after each query. The sum of the matrix after all queries are applied is 23.
```

Example 2:

Initial Matrix

0	0	0
0	0	0
0	0	0

Query 0

4	4	4
0	0	0
0	0	0

Query 1

4	4	4
2	2	2
0	0	0

Query 2

1	4	4
1	2	2
1	0	0

Query 3

1	4	4
1	2	2
3	3	3

Query 4

1	4	1
1	2	1
3	3	1

```
Input: n = 3, queries = [[0,0,4],[0,1,2],[1,0,1],[0,2,3],[1,2,1]]
Output: 17
Explanation: The image above describes the matrix after each query. The sum of the matrix after all queries are applied is 17.
```

Constraints:

- `1 <= n <= 10^4`
- `1 <= queries.length <= 5 * 10^4`
- `queries[i].length == 3`
- `0 <= type_i <= 1`
- `0 <= index_i < n`
- `0 <= val_i <= 10^5`

JavaScript

```
const matrixSumQueries = (n, queries) => {
    let usedRow = new Set(), usedCol = new Set(), res = 0;
    queries.reverse();
```

```
    for (const [type, idx, v] of queries) {
        if (type == 0) {
            if (!usedRow.has(idx)) {
                usedRow.add(idx);
                let cnt = n - usedCol.size;
                res += v * cnt;
            }
        } else {
            if (!usedCol.has(idx)) {
                usedCol.add(idx)
                let cnt = n - usedRow.size;
                res += v * cnt;
            }
        }
    }
    return res;
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/963424818/) | More Details > (/submissions/detail/963424818/)

Share your acceptance!