Day 15

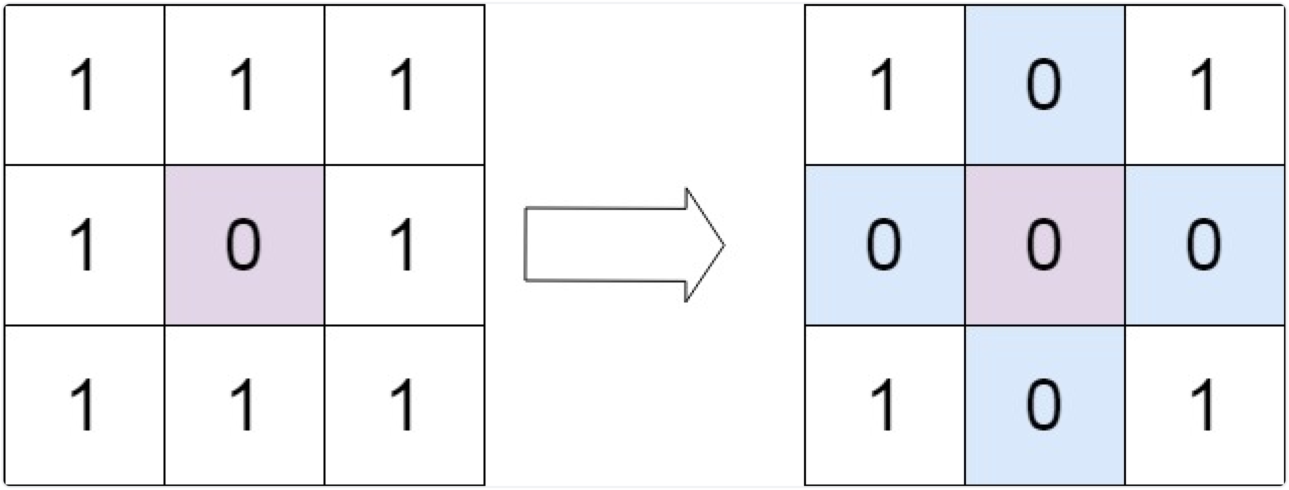
Prepared By Azan Imtiaz

Set Matrix Zeroes (Leatcode)

Given an `m x n` integer matrix `matrix`, if an element is `0`, set its entire row and column to `0`'s.

You must do it in place.

Example 1:

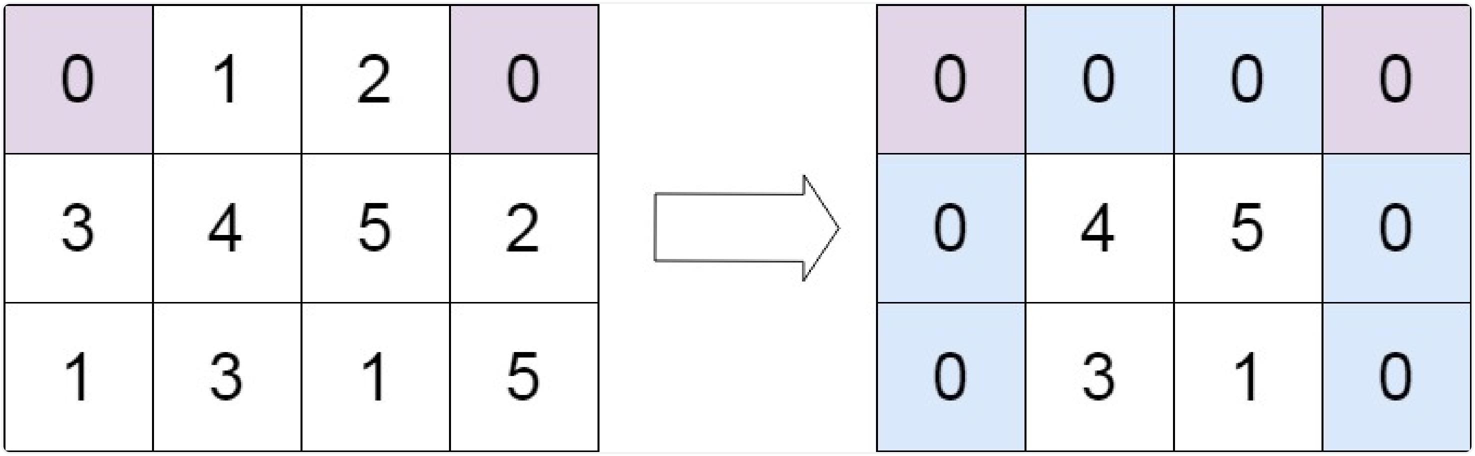


1	1	1
1	0	1
1	1	1

1	0	1
0	0	0
1	0	1

```
Input: matrix = [[1,1,1],[1,0,1],[1,1,1]]
Output: [[1,0,1],[0,0,0],[1,0,1]]
```

Example 2:



0	1	2	0
3	4	5	2
1	3	1	5

0	0	0	0
0	4	5	0
0	3	1	0

```
Input: matrix = [[0,1,2,0],[3,4,5,2],[1,3,1,5]]
Output: [[0,0,0,0],[0,4,5,0],[0,3,1,0]]
```

Constraints:

- `m == matrix.length`
- `n == matrix[0].length`
- `1 <= m, n <= 200`
- -2^{31} `<= matrix[i][j] <=` $2^{31} - 1$

Matrix Zeroes Intuition

To set the entire row and column to zeros without using extra space, we can use the first row and the first column of the matrix as markers. These markers will indicate which rows and columns should be set to zero.

Approach

1. **Check First Row and Column:**
   - Determine if the first row or first column contains any zeros. This is important because we'll use these as markers for other rows and columns.
2. **Mark Zeros:**
   - Iterate through the rest of the matrix (excluding the first row and column). If an element is zero, mark its corresponding position in the first row and first column by setting those elements to zero.
3. **Set Zeros Based on Markers:**
   - Use the marked first row and column to set the rest of the matrix's rows and columns to zero.
4. **Handle First Row and Column:**
   - If the first row or column originally contained zeros, set their entire row or column to zero.

Solution

```
class Solution {
    public void setZeroes(int[][] arr) {
        Boolean firstRow = false;
        Boolean firstColumn = false;

        // Step 1: Check first row and column
        for (int i = 0; i < arr.length; i++) {
            for (int j = 0; j < arr[0].length; j++) {
                if (arr[i][j] == 0) {
                    if (i == 0) firstRow = true;
                    if (j == 0) firstColumn = true;
                    arr[i][0] = 0;
                    arr[0][j] = 0;
                }
            }
        }

        // Step 2: Set zeros based on markers
        for (int i = 1; i < arr.length; i++) {
            for (int j = 1; j < arr[0].length; j++) {
                if (arr[i][0] == 0 || arr[0][j] == 0) {
                    arr[i][j] = 0;
                }
            }
        }

        // Step 3: Handle first row
        if (firstRow) {
            for (int j = 0; j < arr[0].length; j++) {
                arr[0][j] = 0;
            }
        }

        // Step 4: Handle first column
        if (firstColumn) {
            for (int i = 0; i < arr.length; i++) {
                arr[i][0] = 0;
            }
        }
    }
}
```

Explanation

- We first check if there are any zeros in the first row or column and remember this.
- Then, we iterate over the matrix starting from index [1][1], marking the first row and column with zeros where needed.
- After marking, we go through the matrix again to set the cells to zero if their corresponding first row or column cell is zero.
- Finally, we handle the first row and column based on our initial check.

Dry Run of the Solution

Test Case 1:

Input: `[[1,1,1],[1,0,1],[1,1,1]]`

Initial State:

- No zeros in the first row or column.
- Matrix:

```
1 1 1
1 0 1
1 1 1
```

Marking Zeros:

- We find a zero at position [1][1].
- Mark the first row and column at indices [1][0] and [0][1] with zero.
- Matrix after marking:

```
1 0 1
0 0 1
1 1 1
```

Setting Rows and Columns to Zero:

- Check each cell excluding the first row and column.
- For cell [1][2], since arr[1][0] is zero, set arr[1][2] to zero.
- For cell [2][1], since arr[0][1] is zero, set arr[2][1] to zero.

- Matrix after setting rows and columns:

```
1 0 1
0 0 0
1 0 1
```

Final Output:

```
1 0 1
0 0 0
1 0 1
```

Test Case 2:

Input: `[[0,1,2,0],[3,4,5,2],[1,3,1,5]]`

Initial State:

- Zeros present in the first row and column.
- Matrix:

```
0 1 2 0
3 4 5 2
1 3 1 5
```

Marking Zeros:

- First row and column already contain zeros, so we remember this for later.
- No other cells in the first row and column need marking.

Setting Rows and Columns to Zero:

- Check each cell excluding the first row and column.
- Since the first row and column are used as markers, we don't change them yet.
- Proceed through the matrix:
  - For cell [1][1], no marker, so it stays the same.
  - For cell [1][2], no marker, so it stays the same.
  - For cell [1][3], no marker, so it stays the same.
  - For cell [2][1], no marker, so it stays the same.

  - For cell [2][2], no marker, so it stays the same.
  - For cell [2][3], no marker, so it stays the same.
- The rest of the matrix remains unchanged because there were no additional zeros found outside the first row and column.

Handling First Row and Column:

- Set the entire first row to zero because `firstRow` is true.
- Set the entire first column to zero because `firstColumn` is true.
- Matrix after handling first row and column:

```
0 0 0 0
0 4 5 2
0 3 1 5
```

Final Output:

- Now we apply the markers from the first row and column to the rest of the matrix.
- For cell [1][1], since arr[0][1] is zero, set arr[1][1] to zero.
- For cell [1][2], since arr[0][2] is not zero, arr[1][2] stays the same.
- For cell [1][3], since arr[0][3] is zero, set arr[1][3] to zero.
- For cell [2][1], since arr[0][1] is zero, set arr[2][1] to zero.
- For cell [2][2], since arr[0][2] is not zero, arr[2][2] stays the same.
- For cell [2][3], since arr[0][3] is zero, set arr[2][3] to zero.
- Final matrix:

```
0 0 0 0
0 4 5 0
0 3 1 0
```

Thank You So Much For Reading