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

You must do it [in place](https://en.wikipedia.org/wiki/In-place_algorithm).

**Example 1:**



**Input:** matrix = [[1,1,1],[1,0,1],[1,1,1]]

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

**Example 2:**



**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]]

**Solution:**

class Solution {

public void setZeroes(int[][] matrix) {

Boolean isCol = false;

int R = matrix.length;

int C = matrix[0].length;

for (int i = 0; i < R; i++) {

// Since first cell for both first row and first column is the same i.e. matrix[0][0]

// We can use an additional variable for either the first row/column.

// For this solution we are using an additional variable for the first column

// and using matrix[0][0] for the first row.

if (matrix[i][0] == 0) {

isCol = true;

}

for (int j = 1; j < C; j++) {

// If an element is zero, we set the first element of the corresponding row and column to 0

if (matrix[i][j] == 0) {

matrix[0][j] = 0;

matrix[i][0] = 0;

}

}

}

// Iterate over the array once again and using the first row and first column, update the elements.

for (int i = 1; i < R; i++) {

for (int j = 1; j < C; j++) {

if (matrix[i][0] == 0 || matrix[0][j] == 0) {

matrix[i][j] = 0;

}

}

}

// See if the first row needs to be set to zero as well

if (matrix[0][0] == 0) {

for (int j = 0; j < C; j++) {

matrix[0][j] = 0;

}

}

// See if the first column needs to be set to zero as well

if (isCol) {

for (int i = 0; i < R; i++) {

matrix[i][0] = 0;

}

}

}

}

**Complexity Analysis**

* Time Complexity : O(M \times N)*O*(*M*×*N*)
* Space Complexity : O(1)*O*(1)