Set Matrix Zeroes

Given a *m* x *n* matrix, if an element is 0, set its entire row and column to 0. Do it in place.

click to show follow up.

Follow up:
Did you use extra space?
A straight forward solution using O(*mn*) space is probably a bad idea.
A simple improvement uses O(*m* + *n*) space, but still not the best solution.
Could you devise a constant space solution?

Solution 1

My idea is simple: store states of each row in the first of that row, and store states of each column in the first of that column. Because the state of row0 and the state of column0 would occupy the same cell, I let it be the state of row0, and use another variable "col0" for column0. In the first phase, use matrix elements to set states in a top-down way. In the second phase, use states to set matrix elements in a bottom-up way.

```
void setZeroes(vector<vector<int> > &matrix) {
    int col0 = 1, rows = matrix.size(), cols = matrix[0].size();

    for (int i = 0; i < rows; i++) {
        if (matrix[i][0] == 0) col0 = 0;
        for (int j = 1; j < cols; j++)
            if (matrix[i][j] == 0)
                matrix[i][0] = matrix[0][j] = 0;
    }

    for (int i = rows - 1; i >= 0; i--) {
        for (int j = cols - 1; j >= 1; j--)
            if (matrix[i][0] == 0 || matrix[0][j] == 0)
                matrix[i][j] = 0;
        if (col0 == 0) matrix[i][0] = 0;
    }
}
```

written by mzchen original link here

Solution 2

```
public class Solution {
public void setZeroes(int[][] matrix) {
    boolean fr = false,fc = false;
    for(int i = 0; i < matrix.length; i++) {
        for(int j = 0; j < matrix[0].length; j++) {
            if(matrix[i][j] == 0) {
                if(i == 0) fr = true;
                if(j == 0) fc = true;
                matrix[0][j] = 0;
                matrix[i][0] = 0;
            }
        }
    }
    for(int i = 1; i < matrix.length; i++) {
        for(int j = 1; j < matrix[0].length; j++) {
            if(matrix[i][0] == 0 || matrix[0][j] == 0) {
                matrix[i][j] = 0;
            }
        }
    }
    if(fr) {
        for(int j = 0; j < matrix[0].length; j++) {
            matrix[0][j] = 0;
        }
    }
    if(fc) {
        for(int i = 0; i < matrix.length; i++) {
            matrix[i][0] = 0;
        }
    }

}
```

}

written by lz2343 original link here

Solution 3

I find the last row which has 0, and use it to store the 0-collumns. Then go row by row set them to 0. Then go column by column set them to 0. Finally set the last row which has 0. It's long but hey it's O(1)

```cpp
class Solution {
public:
    void setZeroes(vector<vector<int> > &matrix) {

        int H = matrix.size();
        int W = matrix[0].size();

        // find the last 0 row
        int last_0_row = -1;
        for (int y = H - 1; y >= 0 && last_0_row == -1; y--)
            for (int x = 0; x < W; x++)
                if (matrix[y][x] == 0)
                {
                    last_0_row = y;
                    break;
                }
        if (last_0_row == -1)
            return;

        // go row by row
        for (int y = 0; y < last_0_row; y++)
        {
            bool this_is_a_0_row = false;

            for (int x = 0; x < W; x++)
            {
                if (matrix[y][x] == 0)
                {
                    this_is_a_0_row = true;
                    matrix[last_0_row][x] = 0;
                }
            }

            if (this_is_a_0_row)
            for (int x = 0; x < W; x++)
            {
                matrix[y][x] = 0;
            }
        }

        // set collums to 0
        for (int y = 0; y < H; y++)
        for (int x = 0; x < W; x++)
        {
            if (matrix[last_0_row][x] == 0)
                matrix[y][x] = 0;
        }

        // set the last 0 row
        for (int x = 0; x < W; x++)
        {
            matrix[last_0_row][x] = 0;
        }
    }
};
```

written by lugiavn original link here

From Leetcoder.