

Ash Goyal challenge 45 - 60 days

Set matrix zeroes

73. Set Matrix Zeroes

Medium

11.4K

621



Companies

Given an $m \times n$ 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	0	1		0	0	0
1	1	1		1	0	1

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

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

where ever we found 0 in the matrix then column and Row of that 0 will convert to 0

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Approach 1: (Brute force)

- 1) First we will traverse the matrix using 2 loops and check if we got zero
- 2) If we got zero then mark all cell in row i and column j with -1 except 0
- 3) Then Run another loop to change value of -1 to 0 so that we avoid other rows and column for being converted

Q why we are not making it zero?

→ So consider this example we will mark every row and column 0 at required coordinates.

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

at index $(0,0)=1$, $(0,1)=1$, $(0,2)=1$; $(0,3)=1$, $(1,0)=1$

at index $(1,1)=0$

||

convert whole Row & column to 0

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

at index $(1,2)=0$ (again)

at index $(1,3)=0$

$$\begin{array}{c} 0 \\ 1 \\ 2 \end{array} \begin{array}{c} 0 \\ \downarrow \\ 1 \\ 2 \end{array} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Means everything become zero so we need another Number to replace

Code :-

```
void setMat (vector<vector<int>>& matrix) {
```

```
    int n = matrix.size() // Rows
```

```
    int m = matrix[0].size() // Columns
```

```
    for (Row i) // Rows
```

```
        for (Column j) // Cols
```

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

```
                markCol (matrix, j, n)
```

```
                markRow (matrix, i, m)
```

```
            }
```

```
        }
```

```
    }
```

```
void markCol (vector<vector<int>> matrix, int j, int n)
```

```
    for (int k=0; k<n; k++) {
```

```
        if (matrix[k][j] != 0)
```

```
            matrix[k][j] = -1;
```

```
        }
```

```
    }
```

```
void markRow(vector<vector<int>>& matrix, int i, int m) {
```

```
    for (int k=0; k<m; k++)
```

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

```
            matrix[i][k] = -1;
```

```
        }
```

```
    }
```

```
}
```

$T.C = O(n \times m)^3$

```
for (row(i))
```

```
    for (col(j))
```

```
        if (matrix[i][j] == -1)
```

```
            matrix[i][j] = 0
```

```
        }
```

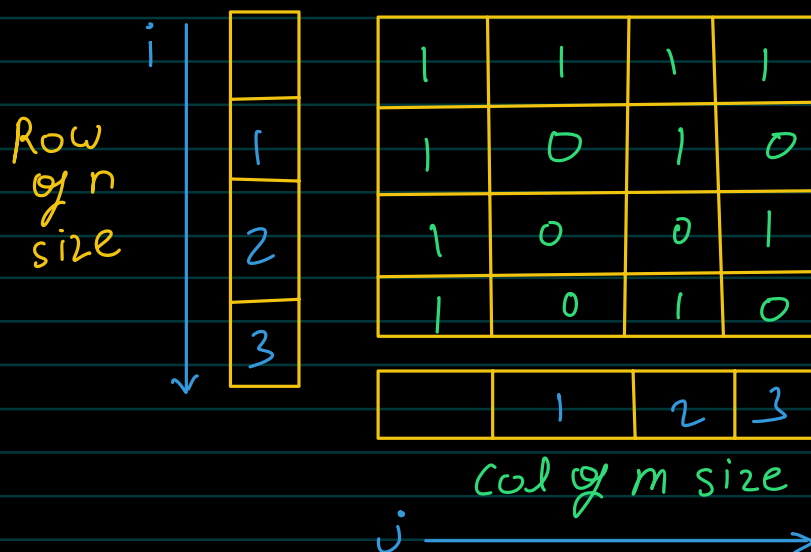
```
    }
```

```
}
```

$S.C = O(1)$

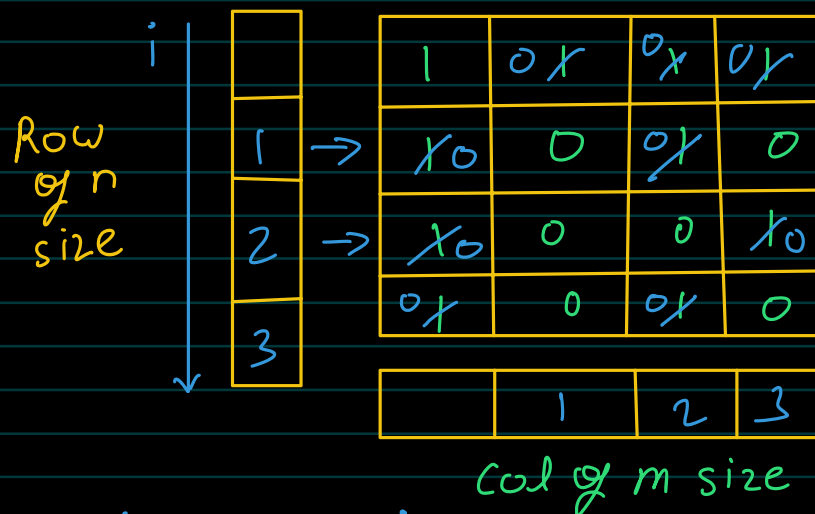
Better approach: (2 Set approach)

This approach we will create 2 unordered_set which will keep the track of zeros in column. & row



Traverse through the matrix and if we encounter zero then mark Row and column with the value of i of the loop

$T.C = O(m \times n)$



Again iterate using 2 loop and check if Row[i] count is than more 0 or col[j] is more than 0 if Yes Set matrix to the 0

T.C - $O(m \times n)$
S.C - $O(n) + O(m)$

1	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

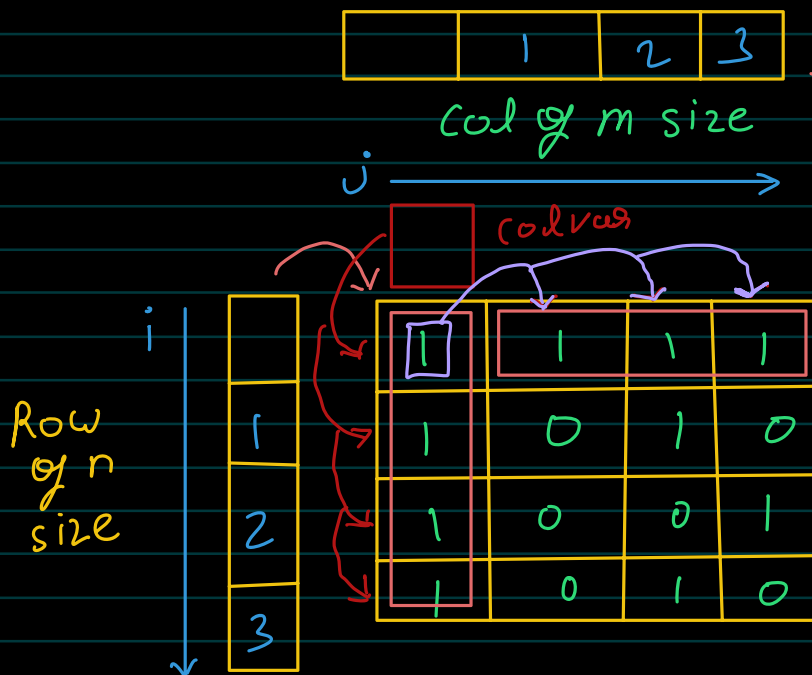
Result

$$T.C \rightarrow (2 \times m \times n)$$

$$S.C \rightarrow O(n) + O(m)$$

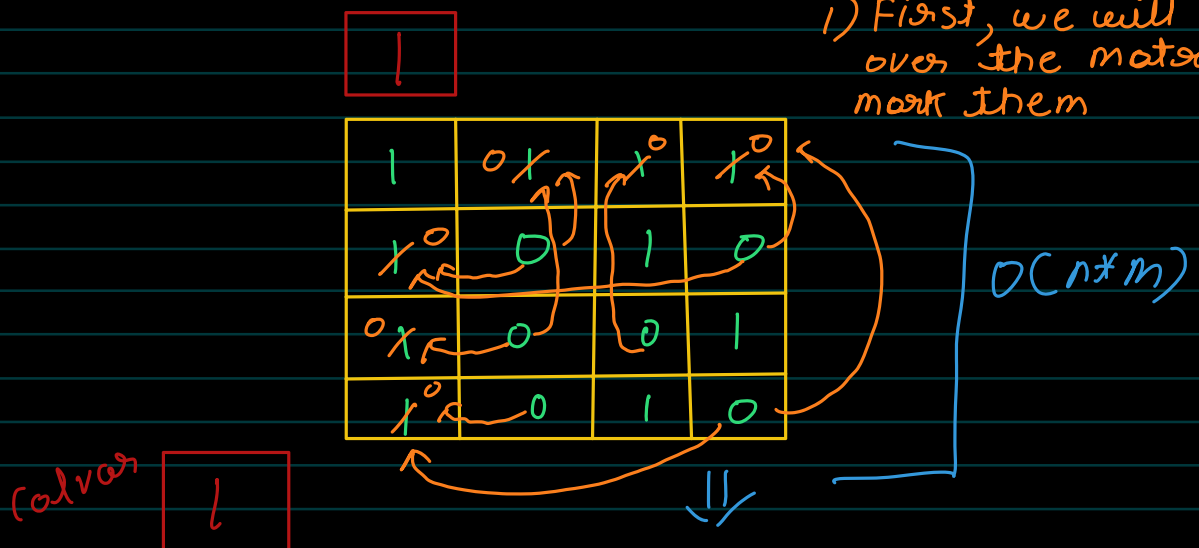
Optimizing Space:-

To optimize the space we are going to use matrix itself to track zero.

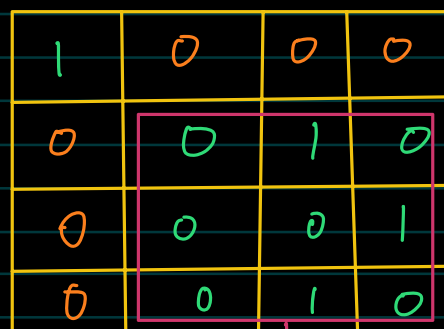


If matrix[0][0] == 0 then whole column will be 0

If colvar is 0 then whole Row will be 0



1) First, we will iterate over the matrix mark them



We will not modify the 1st row and column, we will modify matrix from (1,1) to (n-1, m-1)

Traversal Matrix

if we found anything != 0

and matrix[0][i] and matrix[j][0] is 0 then set matrix to zero

1			
1	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Now if colvar is 0 then Run a loop and mark whole Row to 0 $matrix[i][0]$

If $matrix[0][0]$ is 0 then mark $matrix[0][j]$

The steps are as follows:

1. First, we will traverse the matrix and mark the proper cells of 1st row and 1st column with 0 accordingly. The marking will be like this: if cell (i, j) contains 0, we will mark the i -th row i.e. $matrix[i][0]$ with 0 and we will mark j -th column i.e. $matrix[0][j]$ with 0.

If i is 0, we will mark $matrix[0][0]$ with 0 but if j is 0, we will mark the `col0` variable with 0 instead of marking $matrix[0][0]$ again.

2. After step 1 is completed, we will modify the cells from $(1,1)$ to $(n-1, m-1)$ using the values from the 1st row, 1st column, and `col0` variable.

We will not modify the 1st row and 1st column of the matrix here as the modification of the rest of the matrix i.e. From $(1,1)$ to $(n-1, m-1)$ is dependent on that row and column.

3. Finally, we will change the 1st row and column using the values from $matrix[0][0]$ and `col0` variable. Here also we will change the row first and then the column.

If $matrix[0][0] = 0$, we will change all the elements from the cell $(0,1)$ to $(0, m-1)$, to 0.

If `col0` = 0, we will change all the elements from the cell $(0,0)$ to $(n-1, 0)$, to 0.

Observations: Why in the second step, we are first marking the matrix from the cell $(1,1)$ to $(n-1, m-1)$ and not from $(0,0)$:

Let's understand this using the following example:

Given matrix:

1	1	1	1
1	0	1	1
1	1	0	1
0	1	1	1

Now, we will try to apply step 1 in the above matrix, `col0` will be 0 as $(3,0)$ contains 0 and it will look like the following:

1	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1

Now, in the second step we will try to start modifying the cells with value 0 from $(0,0)$. First, we will change the value of $(0,0)$ to 0 as `col0` is marked with 0. After that, while checking for cell $(0, 3)$ we will find that the value of $(0,0)$ is 0. And we will again modify the cell $(0,3)$ with 0. But this should not happen as $(0,0)$ was initially 1 and that is why

Taken
Take you
- forward
for Better
Understanding

Code Brute force

```
class Solution {
public:
    void MarkCol(vector<vector<int>>& matrix,int j,int m){
        for(int k=0;k<m;k++){
            if(matrix[k][j]!=0)
            {
                matrix[k][j]=-2147483602;
            }
        }
    }
    void Markrow(vector<vector<int>>& matrix,int i,int n){
        for(int j=0;j<n;j++){
            if(matrix[i][j]!=0)
            {
                matrix[i][j]=-2147483602;
            }
        }
    }
    void setZeroes(vector<vector<int>>& matrix) {
        int n=matrix.size();//row
        int m=matrix[0].size();//col
        for(int i=0;i<n;i++){
            for(int j=0;j<m;j++){
                if(matrix[i][j]==0){
                    MarkCol(matrix,j,n);
                    Markrow(matrix,i,m);
                }
            }
        }
        for(int i=0;i<n;i++){
            for(int j=0;j<m;j++){
                if(matrix[i][j]==-2147483602){
                    matrix[i][j]=0;
                }
            }
        }
    }
};
```

Code Better approach

```
class Solution {
public:
    void setZeroes(vector<vector<int>>& matrix) {
        int n=matrix.size();//row
        int m=matrix[0].size();//col
        unordered_set<int> row;
        unordered_set<int> col;

        for(int i=0;i<n;i++){
            for(int j=0;j<m;j++){
                if(matrix[i][j]==0){
                    row.insert(i);
                    col.insert(j);
                }
            }
        }
        for(int i=0;i<n;i++){
            for(int j=0;j<m;j++){
                if(row.count(i)>0 || col.count(j)>0){
                    matrix[i][j]=0;
                }
            }
        }
    }
};
```


Code optimal approach

```
class Solution {
public:
    void setZeroes(vector<vector<int>>& matrix) {
        int n=matrix.size();
        int m=matrix[0].size();
        int Colvar=1;
        for(int i=0;i<n;i++){
            for(int j=0;j<m;j++){
                if(matrix[i][j]==0){
                    matrix[i][0]=0;
                    if(j!=0){
                        matrix[0][j]=0;
                    }
                    else{
                        Colvar=0;
                    }
                }
            }
        }
        for(int i=1;i<n;i++){
            for(int j=1;j<m;j++){
                if(matrix[i][j]!=0){
                    if(matrix[i][0]==0 || matrix[0][j]==0){
                        matrix[i][j]=0;
                    }
                }
            }
        }
        if(matrix[0][0]==0){
            for(int j=0;j<m;j++) matrix[0][j]=0;
        }
        if(Colvar==0){
            for(int i=0;i<n;i++) matrix[i][0]=0;
        }
    }
};
```