

## A Boolean Matrix Question

Given a boolean matrix  $mat[M][N]$  of size  $M \times N$ , modify it such that if a matrix cell  $mat[i][j]$  is 1 (or true) then make all the cells of  $i$ th row and  $j$ th column as 1.

Example 1

The matrix

1 0

0 0

should be changed to following

1 1

1 0

Example 2

The matrix

0 0 0

0 0 1

should be changed to following

0 0 1

1 1 1

Example 3

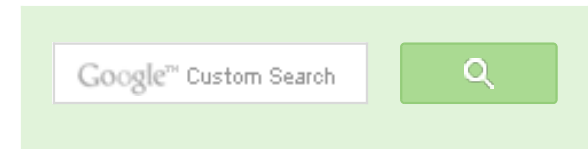
The matrix

1 0 0 1

0 0 1 0

0 0 0 0

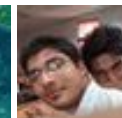
should be changed to following



GeeksforGeeks



53,520 people like [GeeksforGeeks](#).



[Interview Experiences](#)

[Advanced Data Structures](#)

[Dynamic Programming](#)

[Greedy Algorithms](#)

[Backtracking](#)

[Pattern Searching](#)

[Divide & Conquer](#)

[Mathematical Algorithms](#)

[Recursion](#)

[Geometric Algorithms](#)



## Popular Posts

[All permutations of a given string](#)

[Memory Layout of C Programs](#)

[Understanding "extern" keyword in C](#)

[Median of two sorted arrays](#)

[Tree traversal without recursion and without stack!](#)

[Structure Member Alignment, Padding and](#)

[Data Packing](#)

[Intersection point of two Linked Lists](#)

[Lowest Common Ancestor in a BST.](#)

[Check if a binary tree is BST or not](#)

[Sorted Linked List to Balanced BST](#)

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

### Method 1 (Use two temporary arrays)

- 1) Create two temporary arrays row[M] and col[N]. Initialize all values of row[] and col[] as 0.
- 2) Traverse the input matrix mat[M][N]. If you see an entry mat[i][j] as true, then mark row[i] and col[j] as true.
- 3) Traverse the input matrix mat[M][N] again. For each entry mat[i][j], check the values of row[i] and col[j]. If any of the two values (row[i] or col[j]) is true, then mark mat[i][j] as true.

Thanks to [Dixit Sethi](#) for suggesting this method.

```
#include <stdio.h>
#define R 3
#define C 4

void modifyMatrix(bool mat[R][C])
{
    bool row[R];
    bool col[C];

    int i, j;

    /* Initialize all values of row[] as 0 */
    for (i = 0; i < R; i++)
    {
        row[i] = 0;
    }

    /* Initialize all values of col[] as 0 */
    for (i = 0; i < C; i++)
    {
        col[i] = 0;
    }

    /* Store the rows and columns to be marked as 1 in row[] and col[]
       arrays respectively */
    for (i = 0; i < R; i++)
    {
```

```

    for (j = 0; j < C; j++)
    {
        if (mat[i][j] == 1)
        {
            row[i] = 1;
            col[j] = 1;
        }
    }
}

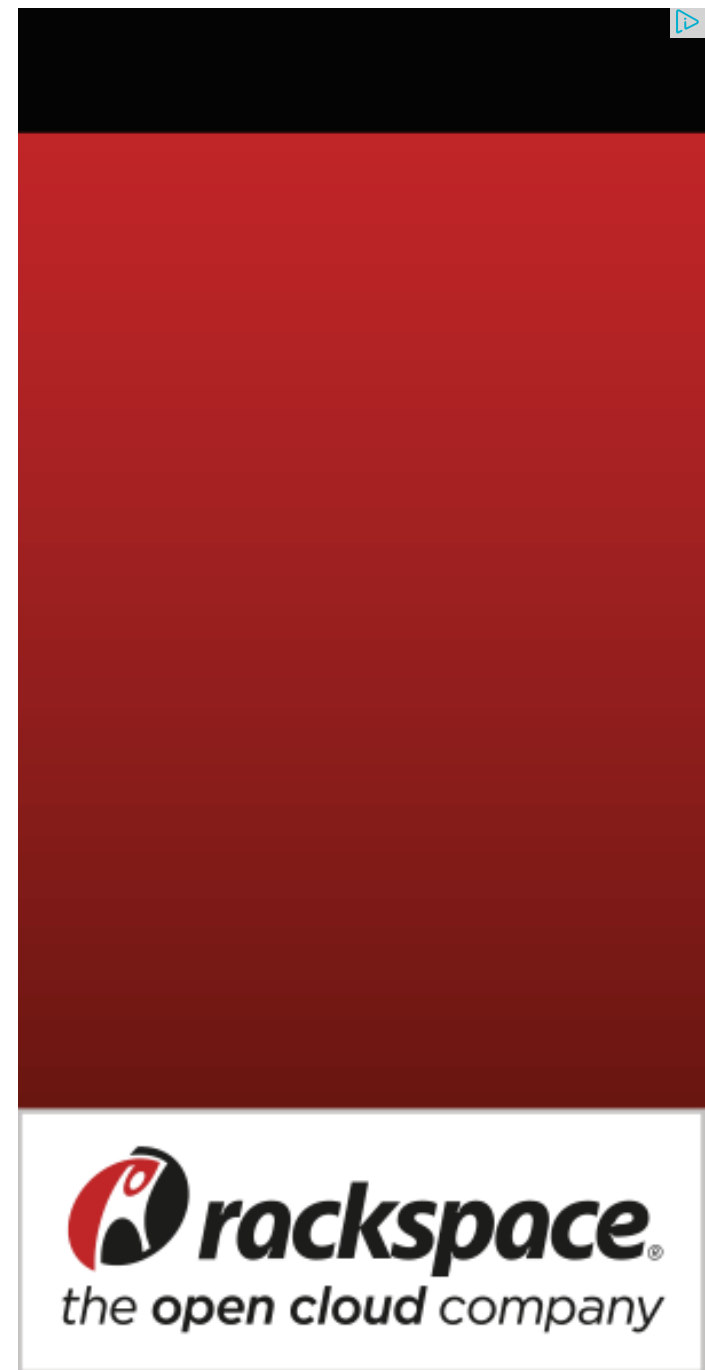
/* Modify the input matrix mat[] using the above constructed row[]
col[] arrays */
for (i = 0; i < R; i++)
{
    for (j = 0; j < C; j++)
    {
        if ( row[i] == 1 || col[j] == 1 )
        {
            mat[i][j] = 1;
        }
    }
}

/* A utility function to print a 2D matrix */
void printMatrix(bool mat[R][C])
{
    int i, j;
    for (i = 0; i < R; i++)
    {
        for (j = 0; j < C; j++)
        {
            printf("%d ", mat[i][j]);
        }
        printf("\n");
    }
}

/* Driver program to test above functions */
int main()
{
    bool mat[R][C] = { {1, 0, 0, 1},
                        {0, 0, 1, 0},
                        {0, 0, 0, 0},
    };

    printf("Input Matrix \n");

```



```

printMatrix(mat);

modifyMatrix(mat);

printf("Matrix after modification \n");
printMatrix(mat);

return 0;
}

```

Output:

```

Input Matrix
1 0 0 1
0 0 1 0
0 0 0 0
Matrix after modification
1 1 1 1
1 1 1 1
1 0 1 1

```

Time Complexity:  $O(M*N)$

Auxiliary Space:  $O(M + N)$

### Method 2 (A Space Optimized Version of Method 1)

This method is a space optimized version of above method 1. This method uses the first row and first column of the input matrix in place of the auxiliary arrays row[] and col[] of method 1. So what we do is: first take care of first row and column and store the info about these two in two flag variables rowFlag and colFlag. Once we have this info, we can use first row and first column as auxiliary arrays and apply method 1 for submatrix (matrix excluding first row and first column) of size  $(M-1)*(N-1)$ .

- 1) Scan the first row and set a variable rowFlag to indicate whether we need to set all 1s in first row or not.
- 2) Scan the first column and set a variable colFlag to indicate whether we need to set all 1s in first column or not.
- 3) Use first row and first column as the auxiliary arrays row[] and col[] respectively, consider the matrix as submatrix starting from second row and second column and apply method 1.

## Recent Comments

**Aman** Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 13 minutes ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 17 minutes ago

**Sanjay Agarwal** bool

tree::Root\_to\_leaf\_path\_given\_sum(tree...

Root to leaf path sum equal to a given number · 42 minutes ago

**GOPI GOPINATH** @admin Highlight this

sentence "We can easily...


Count trailing zeroes in factorial of a number · 44 minutes ago

**newCoder3006** If the array contains negative numbers also. We...

Find subarray with given sum · 1 hour ago

**newCoder3006** Code without using while loop. We can do it...

Find subarray with given sum · 1 hour ago

AdChoices 

[▶ Matrix in Java](#)

[▶ C++ Vector](#)

[▶ Matrix Math](#)

4) Finally, using rowFlag and colFlag, update first row and first column if needed.

Time Complexity:  $O(M*N)$

Auxiliary Space:  $O(1)$

Thanks to [Sidh](#) for suggesting this method.

Please write comments if you find the above codes/algorithms incorrect, or find other ways to solve the same problem.



AdChoices

[▶ Matrix Code](#)

[▶ Matrix Second](#)

[▶ Can Matrix](#)

AdChoices

[▶ Time Matrix](#)

[▶ Matrix 3](#)

[▶ C++ Code](#)

## Related Tpoics:

---

- [Remove minimum elements from either side such that 2\\*min becomes more than max](#)
- [Divide and Conquer | Set 6 \(Search in a Row-wise and Column-wise Sorted 2D Array\)](#)
- [Bucket Sort](#)
- [Kth smallest element in a row-wise and column-wise sorted 2D array | Set 1](#)
- [Find the number of zeroes](#)
- [Find if there is a subarray with 0 sum](#)
- [Divide and Conquer | Set 5 \(Strassen's Matrix Multiplication\)](#)
- [Count all possible groups of size 2 or 3 that have sum as multiple of 3](#)



3



Tweet

0



0

Writing code in comment? Please use [ideone.com](https://ideone.com) and share the link here.

49 Comments

GeeksforGeeks

Sort by Newest ▼



Join the discussion...



**cooldude** • 5 months ago

```
public int[][] changeMatrixRowsColumnAt1(int[][] mat)

{

    boolean fR=false,fC=false;

    for(int col=0;col<mat[0].length;col++) {="" if(mat[0]
```

^ | ▼ • Reply • Share ›



**cooldude** • 5 months ago

```
public int[][] changeMatrixRowsColumnAt1(int[][] mat)

{

    boolean fR=false,fC=false;

    for(int col=0;col<mat[0].length;col++) {="" if(mat[0]
```

^ | ▼ • Reply • Share ›



**cooldude** · 5 months ago

```
public int[][] changeMatrixRowsColumnAt1(int[][] mat)
```

```
{
```

```
    boolean fR=false,fC=false;
```

```
    for(int col=0;col<mat[0].length;col++) {="" if(mat[0][col]=="1") {="" fr="true;" br
    row="0;row<mat[0].length;row++}" {="" if(mat[row][0]=="1") {="" fc="true;" br
    i="1;i<mat.length;i++}" {="" for(int="" j="1;j<mat[0].length;j++)" {="" if(mat[i]
    }="" }="" }="" for(int="" col="1;col<mat[0].length;col++)" {="" if(mat[0][col]=="1"
    row="1;row<mat.length;row++)" {="" mat[row][col]="1;" }="" }="" }="" for(int=""
    {="" if(mat[row][0]=="1") {="" for(int="" col="1;col<mat[0].length;col++)" {="" r
    {="" for(int="" col="0;col<mat[0].length;col++)" mat[0][col]="1;" }="" if(fc)="" {
    row="0;row<mat.length;row++)" mat[row][0]="1;" }="" return="" mat;="" }="">
```

^ | v · Reply · Share ›



**cooldude** · 5 months ago

i think second method is correct as if we are not zeroing the first row or first column variables and when we run our case for submatrix of  $m-1 \times n-1$  then we are only considering first column if in submatrix we have one, if we have zero leave it

counter example given as

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

but as all zero in submatrix first row and column as it is so

final output

```
1 1 1 1
1 1 1 1
```

1 1 1 1

^ | v • Reply • Share ›



**anonymous** • 5 months ago

I have a doubt related to the second method.  
Suppose the matrix is

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

Now both roFlag and colFlag will be set. But how will we know after going through the matrix, which rows and columns have to be set after seeing roFlag and colFlag.

Like in the above example, we have to set two such rows as 1. How is that managed?

^ | v • Reply • Share ›



**Guest** • 5 months ago

Using second method, after preparing first row and first column.  
we can scan the first column and starting from second row, if 1 is found then we have to set the row and column to 1 using memset, that will further improve running time.

1 ^ | v • Reply • Share ›



**Tinku** • 7 months ago

/\* using only one temporary variable, it is done but time complexity is of order O

```
int main()
```

```
{
```

```
int arr[4][4]={1,0,0,0},{0,0,0,0},{0,0,0,0},{0,0,0,0}};
```



```
int arr1[4][4];

int i,j,k;

for(i=0;i<4;i++)

for(j=0;j<4;j++)

arr1[i][j]=0;

for(i=0;i<4;i++)

{
```

[see more](#)

^ | v • Reply • Share ›



**Marsha Donna** • 8 months ago

@GeeksforGeeks i think method 2 wil fail for the following arr

{1,0,0,1

1,0,0,0

1,0,0,0}

because it will give op as

{1,1,1,1

1,0,0,0

1,0,0,0}

whereas corrct op is

{1,1,1,1

1,0,0,1

1,0,0,1}

method2 also requires some auxillary space to hold col indexes of row 0 and r  
that that particular row or col can be modified ...in above eg we need to store a  
col 3 as well...pl correct me if anything is wrong

1 ^ | v • Reply • Share ›



**wasseypuriyan** → Marsha Donna · 7 months ago

We can modify method 2 to first update the submatrix  $(M-1)*(N-1)$  and can modify 1st row and 1st col.

^ | v · Reply · Share ›



**Upasana Sharma** · 8 months ago

Changing element to 2 works for Boolean matrix, and is extendible to any situation of 2.

Geeksforgeeks team, please add!

1 ^ | v · Reply · Share ›



**Guest** · 8 months ago

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
int a[4][4]={
```

```
{1,0,0,1},
```

```
{0,0,0,0},
```

```
{0,0,1,0},
```

```
{0,0,0,0}
```

```
};
```

```
int i,j,x=0,n=4;
```

```
for(i=0;i<n;i++) {="" for(j="0;j<n;j++)" {="" if(a[i][j]=="1)" {="" for(x="0;x<n;x+
```

```
}="" }="" for(x="0;x<=n;x++)" {"="" if(a[x][j]=="0)" {"="" a[x][j]="2;" }="" }="" }="" }="" }=""  
for(j="0;j<=n;j++)" {"="" if(a[i][j]=="2)" a[i][j]="1;" printf("="" %d="" ",a[i][j]);="" }=""  
return="" 0;="" }="">
```

^ | v • Reply • Share ›



**wakeup123** • a year ago

someone Please help me understand the question. I am not getting what is be

^ | v • Reply • Share ›



**Paparao Veeragandham** • a year ago

I think 2nd Method doesn't work.

Example:

Input:

1 0 0 0.

1 0 0 0.

1 0 0 0.

Expecting o/p:

1 1 1 1.

1 1 1 1.

1 1 1 1.

But 2nd Algo Produces:

1 1 1 1.

1 0 0 0.

1 0 0 0.

1 ^ | v • Reply • Share ›



**Himanshu Sardana** • a year ago

Please Explain your Code A lil bit !

^ | v • Reply • Share ›



**Prateek Sharma** · a year ago

Second method fails.....

Consider Example

[1 0 0

0 0 0

1 0 0]

According to Second method ,answer is

[1 1 1

1 0 1

1 0 1]

But the actual answer is

[1 1 1

1 0 0

1 1 1]

According to me problem occurs here:

When we use the first row and first column to store values for rest of matrix, tl column get overwritten which causes problem later. In my example, third row fi

[see more](#)

^ | v · [Reply](#) · [Share](#) ›



**Priyanka** · a year ago

Another solution with more time complexity but constant space complexity

```
void modifyMatrix(bool mat[R][C])
{
    int i, j;

    /* Store the rows and columns to be marked as 1 in row[] and col[]
    arrays respectively */
```

```

for (i = 0; i < R; i++)
{
    for (j = 0; j < C; j++)
    {
        if (mat[i][j] == 1)
        {
            int row=i;
            for(i;i<R;i++)
                mat[i][j]=1;
            for(j;j<C;j++)
                mat[row][j]=1;
        }
    }
}

```

^ | v • Reply • Share ›



**Alien** → Priyanka • 8 months ago

This will not work because there is not way to check that if mat[i][j] is 1 processed or still needs to be process. this will work if we use int matr rows and cols by representing them with 2.

1 ^ | v • Reply • Share ›



**Harshit Gupta** → Priyanka • 8 months ago

will not give correct answer for some cases....

1 ^ | v • Reply • Share ›



**Kartheek J** • a year ago

Please Check Below,I think below is more optimized than method 2

```

void modifyBoolMatrix(int *array,int rowLength,int colLength)

```

```

{

    for (int i= 0 ; i < rowLength; i++)
    {
        for (int j = 0; j < colLength; j++)
        {
            if (*(array+i*colLength)+j) == 1)
            {
                *(array+i*colLength) = 1;
                *(array+j) = 1;
            }
        }
    }

    for (int i= rowLength-1; i >= 0; i--)

```

see more

^ | v • Reply • Share ›



**Ashish Bardhan** • a year ago

I've used another approach using Bitwise operators.

Here's the code with the results.

[http://ideone.com/3E9Yov#li\\_y1...](http://ideone.com/3E9Yov#li_y1...)

Time Complexity :  $O(M*N)$ .

Space Complexity :  $O(1)$ .

^ | v • Reply • Share ›



**Aashish** • 2 years ago

Solution for approach 2

```

void flip(int M[][COL])

```

```

{
    int i, j, r, c, flagRow, flagCol;

    for( r = 0, flagCol = 0; r < ROW; ++r )
        flagCol |= M[r][0];

    for( c = 0, flagRow = 0; c < COL; ++c )
        flagRow |= M[0][c];

    for( r = 1; r < ROW; ++r )
        for( c = 1; c < COL; ++c )
        {
            M[0][c] |= M[r][c];
            M[r][0] |= M[r][c];
        }
}

```

[see more](#)

^ | v • Reply • Share ›



**dreamer** • 2 years ago

An easier method which will work and no extra space will be needed:

Step 1: Scan each row and if any element is 1 then make following changes in

a) If element is 0, change it to 2.

b) If it is 1 or 2, leave it.

Step 2. For the matrix change all elements which have a value 2 to 1.

Time Complexity :  $O(m*n)$ .

No extra space needed.

Please point out any cases for which this might not work.

/\* Paste your code here (You may **delete** these lines **if not** writing c)

3 ^ | v • Reply • Share ›



**Upasana Sharma** → dreamer • 8 months ago

perfect!

But the complexity is in  $O(n*n*n)$

Cube.

^ | v • Reply • Share ›



**shek8034** → dreamer • 11 months ago

Your logic is correct... I was thinking the same logic but then i found yo

^ | v • Reply • Share ›



**rahul sundar** → dreamer • a year ago

Works perfect for non boolean matrix :-)

```
/* Paste your code here (You may delete these lines if not wr
```

^ | v • Reply • Share ›



**shek8034** → rahul sundar • 11 months ago

Your logic is correct... I was thinking the same logic but then i f

^ | v • Reply • Share ›



**rk** → shek8034 • 10 months ago

will u guys mind explaining how complexity is  $O(m*n)$ ?

```
/* Paste your code here (You may delete these li
```

^ | v • Reply • Share ›



**anu** • 2 years ago



The method 2 fails when there is only one 1 in the 1st row and the entire column the 1st column and the entire row is 0's. because we are not storing that info a

ex:

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

^ | v · Reply · Share ›



**vk** → anu · 2 years ago

heu anu,

rowflag is a 1xn and col flag is a mx1 array

thus will work properly..... i think u assumed it to be a single variable (v

^ | v · Reply · Share ›



**anu** → vk · 2 years ago

Then what is the point in optimizing the space if they are arrays

^ | v · Reply · Share ›



**shiv** → anu · 2 years ago

hey vk can u xplain d lgic plz in detail ...method 2

^ | v · Reply · Share ›



**sudoHack** · 2 years ago

Can anyone please explain output for this case acc to 2nd method:

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

As 2nd method says:

step1 : scan row and col and set colflag and rowflag accordingly which in this

row and col is one.

Step2: apply method one on submatrix , then sub matrix will be

1 1 1

1 1 1

1 1 1

step3 : now update row1 and col1 acc to rowflag and colflag (which are 0 ), so

0 0 0 0

0 1 1 1

0 1 1 1

0 1 1 1

but actual ans is :

0 1 1 1

1 1 1 1

0 1 1 1

1 1 1 1

^ | v • Reply • Share ›



**Ashish** • 2 years ago

I think the second method will fail for this example

1 0 0 1

1 0 0 0

1 0 0 0

output will be

1 1 1 1

1 1 1 1

1 1 1 1

Can anyone please confirm.

// Paste your code here (you may ~~delete~~ these lines if not writing c

^ | v • Reply • Share ›



**kartik** → Ashish • 2 years ago

Take a closer look at the problem statement. The following output is cc

1 1 1 1

1 1 1 1

1 1 1 1

^ | v • Reply • Share ›



**Bharti** • 2 years ago

I doubt whether only two variables rowFlag and columnFlag will be able to hold and columns. Anybody please explain how is it possible?

For eg: lets say in first row we have elements 2 and 4th set. Now we know tha how to store information about 2nd and 4th columns?

^ | v • Reply • Share ›



**Bharti** → Bharti • 2 years ago

Its clear to me now. Kindly ignore the post.

^ | v • Reply • Share ›



**naveen** → Bharti • 2 years ago

I am also having the doubt you had but i cant figure out how info out

^ | v • Reply • Share ›



**Dixit Sethi** • 3 years ago

Thanks Sidh. Your idea is really appreciable :)

^ | v • Reply • Share ›



PsychoCoder • 3 years ago

Working code for method 2.

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>

int** allocateMatrix (int row, int col) {
    int **matrix , i;
    matrix = (int ** ) malloc (sizeof(int *) * row) ;
    for ( i = 0 ; i < row ; i ++ )
        *(matrix + i) = (int *) malloc (sizeof(int) * col) ;
    return matrix ;
}

int** getInput (int row, int col) {
    int **matrix , i , j;
    matrix = allocateMatrix (row,col) ;
    for ( i = 0 ; i < row ; i ++ ) {
```

see more

^ | v • Reply • Share ›



Amol → PsychoCoder • 2 years ago

Running solution based on your code..

<http://ideone.com/RdFgi>

^ | v • Reply • Share ›



chaitu2289 • 3 years ago

Code for the second solution.

<http://ideone.com/GGgkq>

^ | v • Reply • Share ›



**CaesiumX** • 3 years ago

<http://ideone.com/UHq6V>

^ | v • Reply • Share ›



**rohit** → CaesiumX • 3 years ago

what is the point of sharing this link? the code has compilation errors.

^ | v • Reply • Share ›



**CaesiumX** → rohit • 3 years ago

because compiler over there - gcc  
tested on - turbo c (working fine)

^ | v • Reply • Share ›



**praveen** • 3 years ago

A simple method, time complexity is  $O((M+N)*M*N)$  in worst case.

Matrix cells for which row and column are 0 are the only cells which are going to be 1. So for every cell `mat[i][j]`, check if `i`th row and `j`th column both are zero. If

^ | v • Reply • Share ›



**devan** • 3 years ago

Method 1 is wrong. If you traverse through the 2D array and change the entries, the matrix will be 1's. We cannot distinguish the 1's that were already present and

^ | v • Reply • Share ›



**GeeksforGeeks** → devan • 3 years ago

@devan: Thanks for pointing this out. We have removed Method 1 from the list. Method 3 is method 2. Keep it up!!

^ | v • Reply • Share ›



harsh · 3 years ago

How about the following question for non-boolean numbers.

If a matrix cell  $\text{mat}[i][j]$  is 0 then make all the cells of  $i$ th row and  $j$ th column as

Ex

7 4 8 5

0 0 6 0

0 4 3 0

should be changed to following

0 0 8 0

0 0 0 0

0 0 0 0

^ | v · Reply · Share ›



rohit · 3 years ago

Method 2 is fantastic :)

1 ^ | v · Reply · Share ›



Subscribe



Add Disqus to your site

@geeksforgeeks, **Some rights reserved**

**Contact Us!**

Powered by **WordPress** & **MooTools**, customized by geeksforgeeks team