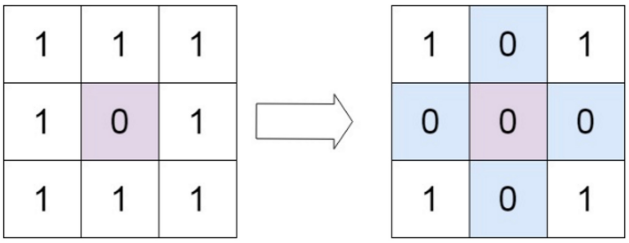
27.Set Matrix Zeroes

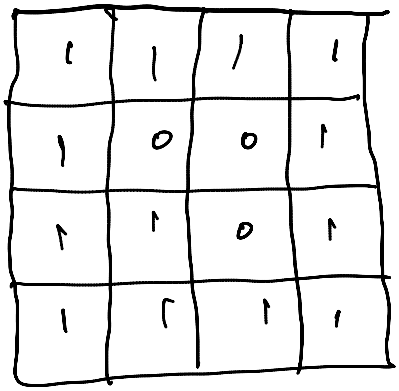
Given an `m x n` integer matrix `matrix`, if an element is `0`, set its entire row and column to `0`'s.

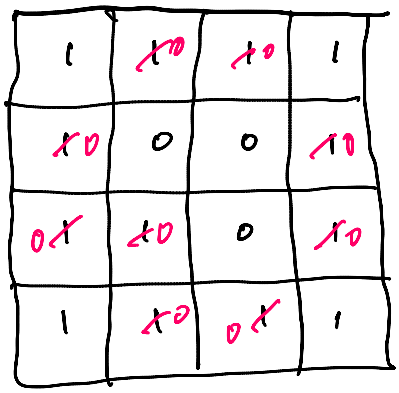
You must do it in place.

Example 1:




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





i) Brute force:

→ Iterate over the matrix

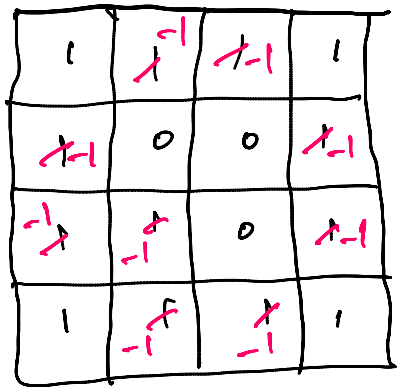
→ if we find '1' don't do anything

– if it's '0' the we mark the row & col as '0'.

BUT It's not correct because array will look different

BUT It's not correct because array will look different and again on next iteration for the next '0' you mark again the row and column will be '0'.

→ Rather go through entire column/row and mark them as '-1'.



→ Now do one more iteration and convert them into zero.

Pseudo Code

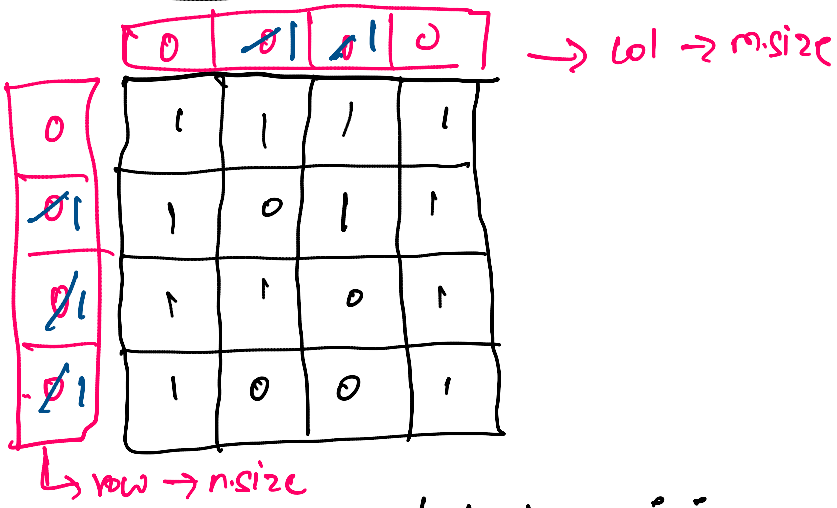
```
for(i=0; i<n; i++)
{
  for(j=0; j<m; j++)                    ] n x m
  {
    if (arr[i][j] == 0)
    {
      markRow(i)  → n
      markCol(j)  → m
    }
```

```
}
markRow(i)   <-- row no
{
  for(j=0; j<m; j++) // Iterate for every col
  {
    if (arr[i][j] != 0)
        arr[i][j] = -1;
  }
}
markCol(j)   <-- col nos
{
  for(i=0; i<n; i++) // Iterate for every row
  {
    if (arr[i][j] != 0)
        arr[i][j] = -1;
  }
}

// Now iterate over the matrix if it's -1 => 0
for(i=0 -> n)
{
  for(j=0 -> m)
  {
    if (arr[i][j] == -1)
        arr[i][j] = 0;
  }
}
```

T.C $\Rightarrow$ $(n \times m) \times (n + m) + (n \times n)$ S.C $\Rightarrow O(1)$

≈ power of cube.

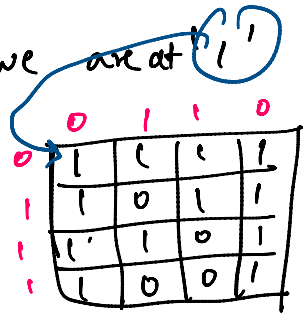
2) Better Solution:



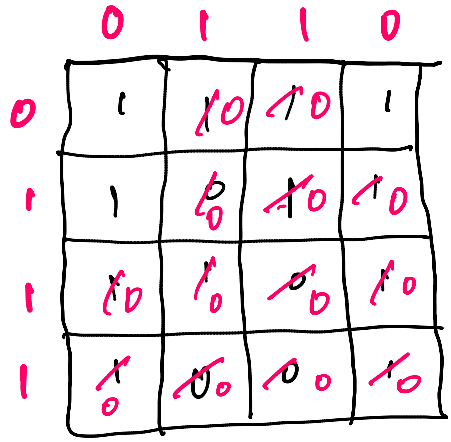
- if any row/col has minimum of one '0', mark the col/row.
- keep a track of '0'. So you can mark the row/col.
- So now there will be four size array for row/col.
- Initially the array value be '0'.
- Start iterating.
  - Whenever '1' ⇒ ignore
    '0' ⇒ mark '1' in the array. for both row/col array.
- At the end we marked the row/col array as '1'.

- Now do a reiteration whenever we are at '1'

will it be '0'? it only be '0' if the arr row/col is marked as '1'.



- If not marked '1' then the original array will be '1' only.
- Do the same for everything



col[m] = {0} row[n] = {0}

```
for (i=0 ⟶ n)
  for (j=0 ⟶ m)
  {
    if (arr[i][j] == 0)
      {  row[i] = 1   // both have been marked '1'.
         col[j] = 1
      }
  }
// Again reiterate. convert 1 → 0
```

for (i = 0 → n)

{ for(j = 0 → n)

{ if (row [i] || col [j])

arr [i] [j] = 0

}

}

Code Studio Solution

```cpp
#include <bits/stdc++.h>
vector<vector<int>> zeroMatrix(vector<vector<int>> &matrix, int n, int m) {
    int col[m] = {0};
    int row[n] = {0};
    for(int i = 0;i<n;i++){
        for(int j=0;j<m;j++){
            if(matrix[i][j] == 0){
                row[i] = 1;
                col[j] = 1;
            }
        }
    }
    for(int i = 0; i<n;i++){
        for(int j=0;j<m;j++){
            if(row[i] || col[j]){
                matrix[i][j] = 0;
            }
        }
    }
    return matrix;
}
```

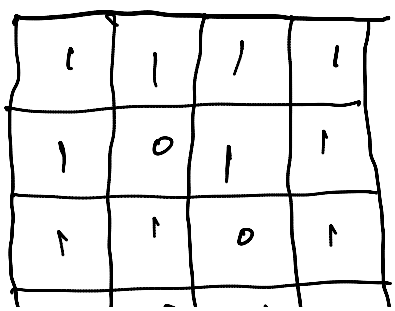
Leet Code Solution

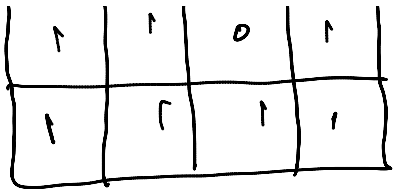
```cpp
class Solution
{
public:
    void setZeroes(vector<vector<int>> &matrix)
    {
        // Declaration and initialization of two integers named rows and cols,
        // where rows is assigned the size of the first dimension of the matrix vector
        // and cols is assigned the size of the second dimension of the matrix vector.
        int rows = matrix.size(), cols = matrix[0].size();

        // Declaration and initialization of two vectors of integers
        // named dummy1 and dummy2, both are assigned to a size equal to the size of
        // the respective dimensions of the matrix vector, and each element of these vectors
        // is assigned the value -1.
        vector<int> dummy1(rows, -1), dummy2(cols, -1);

        // A loop that iterates over the rows and columns of the matrix vector.
        // If the current element of the matrix vector is equal to 0, then
        // the corresponding element of the dummy1 vector is set to 0,
        // and the corresponding element of the dummy2 vector is also set to 0.
        for (int i = 0; i < rows; i++)
        {
            for (int j = 0; j < cols; j++)
            {
                if (matrix[i][j] == 0)
                {
                    dummy1[i] = 0;
                    dummy2[j] = 0;
                }
            }
        }

        // A loop that iterates over the rows and columns of the matrix vector.
        // If the current element of the dummy1 vector is equal to 0, or
        // the current element of the dummy2 vector is equal to 0, then
        // the corresponding element of the matrix vector is set to 0.
        for (int i = 0; i < rows; i++)
        {
            for (int j = 0; j < cols; j++)
            {
                if (dummy1[i] == 0 || dummy2[j] == 0)
                {
                    matrix[i][j] = 0;
                }
            }
        }
    }
};
```

T.C ⇒ $O(2 * n * m)$

S.C ⇒ $O(n) + O(m)$

3rd Approach





- keep a track in the matrix itself
- consider 1st column to be the row

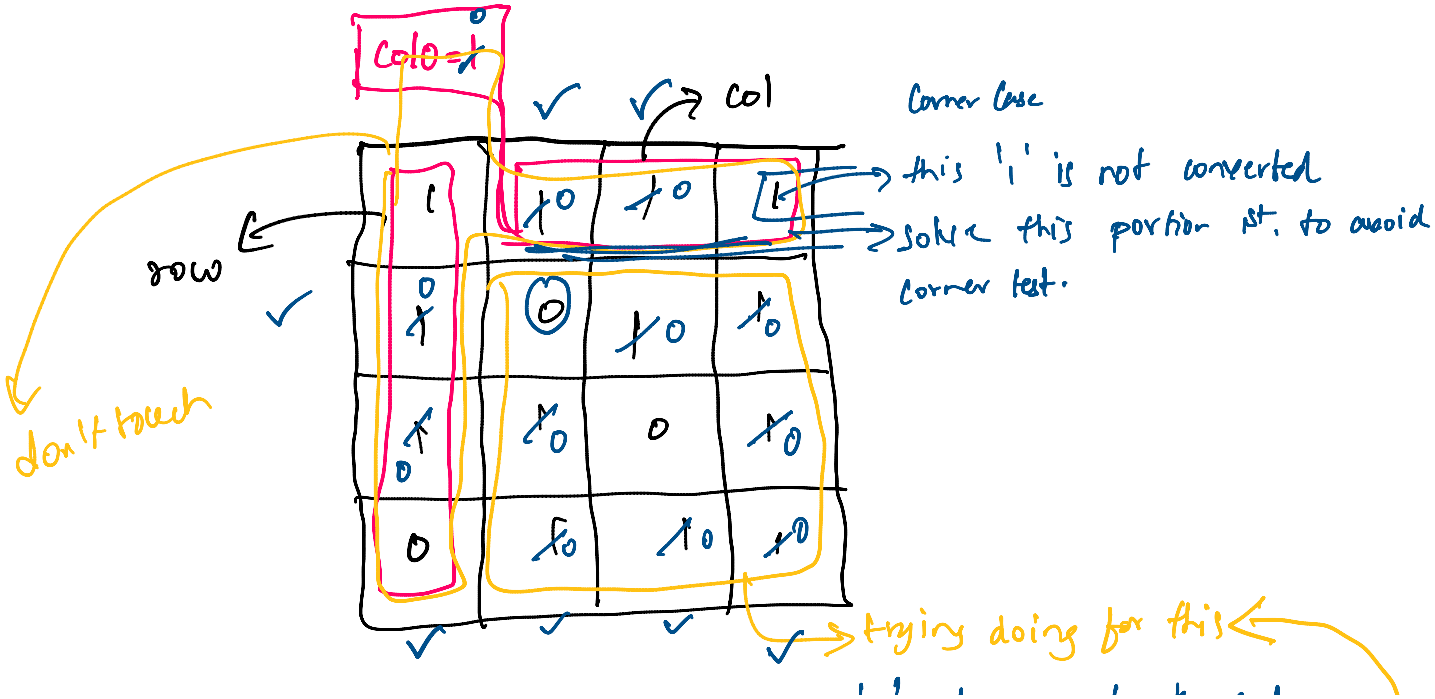
$$col[0] \rightarrow row[r]$$

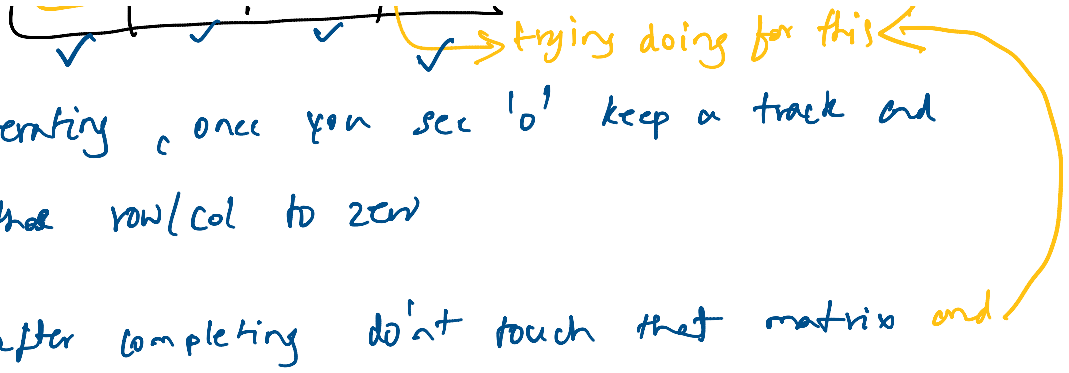
- same 1st row to be the col

$$row[0] \rightarrow col[r]$$

But, it's NOT correct since matrix[0][0] is overlapping

So, take seperate variable colo, now there won't be any collision.





T.C => O(n*m) S.C => O(1)

```cpp
#include <bits/stdc++.h>
vector<vector<int>> zeroMatrix(vector<vector<int>> &matrix, int n, int m) {
    // int col[m] = {0}; -> matrix[0][..]
    // int row[n] = {0}; -> matrix[..][0]
    int col0 = 1;
    for(int i = 0;i<n;i++){
        for(int j=0;j<m;j++){
            if(matrix[i][j] == 0){
                //mark the i-th row
                matrix[i][0] = 0;
                //mark the j-th row
                if(j!= 0){
                matrix[0][j] = 0;
                } else {
                    col0 = 0;
                }

            }
        }
    }
    for(int i = 1;i<n;i++){
        for(int j=1;j<m;j++){
            if(matrix[i][j] != 0){
                // check for the col and row
                if(matrix[0][j] == 0 || matrix[i][0] == 0){
                    matrix[i][j] = 0;
                }
            }
        }
    }
    if(matrix[0][0] == 0){
        for(int j = 0;j<m;j++) matrix[0][j] = 0;
    }
    if (col0 == 0) {
        for(int i = 0;i<n;i++)
            matrix[i][0] = 0;
    }
    return matrix;
}
```