**Boolean Matrix: -**

**Medium** Accuracy: **40.05%** Submissions: **100K+** Points: **4**

Given a boolean matrix of size RxC where each cell contains either 0 or 1, modify it such that if a matrix cell matrix[i][j] is 1 then all the cells in its ith row and jth column will become 1.

**Example 1:**

**Input:**

R = 2, C = 2

matrix[][] = {{1, 0},

{0, 0}}

**Output:**

1 1

1 0

**Explanation:**

Only cell that has 1 is at (0,0) so all

cells in row 0 are modified to 1 and all

cells in column 0 are modified to 1.

**Example 2:**

**Input:**

R = 4, C = 3

matrix[][] = {{ 1, 0, 0},

{ 1, 0, 0},

{ 1, 0, 0},

{ 0, 0, 0}}

**Output:**

1 1 1

1 1 1

1 1 1

1 0 0

**Explanation:**

The position of cells that have 1 in

the original matrix are (0,0), (1,0)

and (2,0). Therefore, all cells in row

0,1,2 are and column 0 are modified to 1.

**Your Task:**  
You dont need to read input or print anything. Complete the function **booleanMatrix()** that takes the matrix as input parameter and modifies it **in-place**.

**Expected Time Complexity:** O(R \* C)  
**Expected Auxiliary Space:** O(R + C)

**Constraints:**  
1 ≤ R, C ≤ 1000  
0 ≤ matrix[i][j] ≤ 1

**Code: -**

//{ Driver Code Starts

#include <bits/stdc++.h>

using namespace std;

// } Driver Code Ends

class Solution

{

public:

//Function to modify the matrix such that if a matrix cell matrix[i][j]

//is 1 then all the cells in its ith row and jth column will become 1.

void booleanMatrix(vector<vector<int> > &matrix){

unordered\_map<int,bool> row, col;

int n = matrix.size(), m = matrix[0].size();

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

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

if(matrix[i][j]==1){

row[i] = true;

col[j] = true;

}

}

}

// row change

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

if(row[i] == true){

for(auto &item:matrix[i])

item = 1;

}

}

// col change

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

if(col[j] == true){

for(int i=0; i<n; ++i)

matrix[i][j] = 1;

}

}

return;

}

};

//{ Driver Code Starts.

int main() {

int t;

cin>>t;

while(t--)

{

int row, col;

cin>> row>> col;

vector<vector<int> > matrix(row);

for(int i=0; i<row; i++)

{

matrix[i].assign(col, 0);

for( int j=0; j<col; j++)

{

cin>>matrix[i][j];

}

}

Solution ob;

ob.booleanMatrix(matrix);

for (int i = 0; i < row; ++i)

{

for (int j = 0; j < col; ++j)

{

cout<<matrix[i][j]<<" ";

}

cout<<endl;

}

}

return 0;

}

// } Driver Code Ends

**T.C: - O(R \* C)**

**S.C: - O(R + C)**