**Unique rows in boolean matrix :-**

Easy Accuracy: 48.36% Submissions: 32K+ Points: 2

Given a binary matrix your task is to find all unique rows of the given matrix in the order of their appearance in the matrix.

**Example 1:**

**Input:**

row = 3, col = 4

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

**Output:** $1 1 0 1 $1 0 0 1 $

**Explanation:** Above the matrix of size 3x4

looks like

1 1 0 1

1 0 0 1

1 1 0 1

The two unique rows are R1: {1 1 0 1} and R2: {1 0 0 1}.   
As R1 first appeared at row-0 and R2 appeared at row-1, in the resulting list, R1 is kept before R2.

**Example 2:**

**Input:**

row = 2, col = 4

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

**Output: $**0 0 0 1 $

**Explanation:** Above the matrix of size 2x4

looks like

0 0 0 1

0 0 0 1

Only unique row is $0 0 0 1 $

**Your Task:**  
You only need to implement the given function **uniqueRow()**. The function takes three arguments the first argument is a matrix **M** and the next two arguments are **row** and **col** denoting the rows and columns of the matrix. The function should **return** the list of the unique row of the matrix. Do not read input, instead use the arguments given in the function.

**Note:**The driver code prints the rows "$" separated. You have to just return list of rows, you do not have to worry about printing anything.

**Expected Time Complexity:** O(row \* col)  
**Expected Auxiliary Space:** O(row \* col)

**Constraints:**  
1<=row,col<=40  
0<=M[][]<=1

**Code** :-

//{ Driver Code Starts

#include<bits/stdc++.h>

using namespace std;

#define MAX 1000

vector<vector<int>> uniqueRow(int M[MAX][MAX],int row,int col);

// } Driver Code Ends

class Solution

{

public:

// #define MAX 1000

vector<vector<int>> uniqueRow(int mat[MAX][MAX],int row,int col)

{

int num;

unordered\_map<int,bool> mp;

vector<vector<int>> ans;

vector<int> help(col);

for(auto i=0; i<row; i++){

num = 0;

for(auto j=0; j<col; j++){

help[j] = mat[i][j];

num = (num\*10) + mat[i][j];

}

if(!mp[num]){

mp[num] = true;

ans.push\_back(help);

}

}

return ans;

}

};

//{ Driver Code Starts.

int main()

{

int T;

cin>>T;

while(T--)

{

int n,m;

cin>>n>>m;

int a[MAX][MAX];

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

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

cin>>a[i][j];

Solution ob;

vector<vector<int>> vec = ob.uniqueRow(a,n,m);

for(int i = 0;i<vec.size();i++){

for(int x : vec[i]){

cout<<x<<" ";

}

cout<<"$";

}

cout<<endl;

}

}

// } Driver Code Ends

**T.C :- O(row \* col)**

**S.C :- O(row)**