**Make Matrix Beautiful :-**

Medium Accuracy: 64.75% Submissions: 22K+ Points: 4

A beautiful matrix is a matrix in which the sum of elements in each row and column is equal. Given a square matrix of size **N**x**N**. Find the minimum number of operation(s) that are required to make the matrix beautiful. In one operation you can **increment** the value of any one **cell by 1**.

**Example 1:**

**Input**:

N = 2

matrix[][] = {{1, 2},

  {3, 4}}

**Output**:   
4

**Explanation**:

Updated matrix:

4 3

3 4

1. Increment value of cell(0, 0) by 3

2. Increment value of cell(0, 1) by 1

Hence total 4 operation are required.

**Example 2:**

**Input**:

N = 3

matrix[][] = {{1, 2, 3},

  {4, 2, 3},

  {3, 2, 1}}

**Output**:   
6

**Explanation**:

Number of operations applied on each cell are as follows:  
1 2 0  
0 0 0  
0 1 2  
Such that all rows and columns have sum of 9.

**Your Task:**  
You don't need to read input or print anything. Complete the function **findMinOpeartion()**that takes **matrix and n** as input **parameters**and returns the **minimum number of operations**.  
  
**Expected Time Complexity:** O(N \* N)  
**Expected Auxiliary Space:** O(N)  
  
**Constraints:**  
1 <= N <= 103  
1 <= matrix[i][j] <= 106

**Code :-**

//{ Driver Code Starts

#include <bits/stdc++.h>

using namespace std;

// } Driver Code Ends

class Solution

{

public:

//Function to find minimum number of operations that are required

//to make the matrix beautiful.

int findMinOpeartion(vector<vector<int> > matrix, int n){

vector<int> rows(n), cols(n);

int maxi = INT\_MIN;

// for row sum

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

int sum=0;

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

sum += matrix[row][col];

rows[row] = sum;

maxi = max(maxi, sum);

}

// for column sum

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

int sum=0;

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

sum += matrix[row][col];

cols[col] = sum;

maxi = max(maxi, sum);

}

// calculating the answer

int ans=0;

for(auto &item : rows)

ans += abs(item - maxi);

return ans;

}

};

//{ Driver Code Starts.

int main()

{

int t;

cin>>t;

while(t--)

{

int n;

cin>>n;

vector<vector<int> > matrix (n,vector<int>(n));

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

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

cin>>matrix[i][j];

Solution ob;

cout << ob.findMinOpeartion(matrix, n) << endl;

}

return 0;

}

// } Driver Code Ends

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

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