PROBLEM

Row with minimum number of 1's

Easy Accuracy: 25.0% Submissions: 37K+ Points: 2

Given a 2D **binary matrix**(1-based indexed) **a** of dimensions **nxm** , determine the **row** that contains the **minimum number** of **1's**.

Note: The matrix contains only **1's** and **0's**. Also, if two or more rows contain the **minimum number** of **1's**, the answer is the **lowest** of those **indices**.

Example 1:

Input:
n = 4,m = 4
a = [[1, 1, 1, 1],
[1, 1, 0, 0],
[0, 0, 1, 1],
[1, 1, 1, 1]]
Output:
2
Explanation:
Rows 2 and 3 contain the minimum number
of 1's(2 each).Since,row 2 is less than row 3.
Thus, the answer is 2.

Example 2:

Input:
n = 3,m = 3
a = [[0, 0, 0],
[0, 0, 0],
[0, 0, 0]]
Output:
1
Explanation:
All the rows contain the same number
of 1's(0 each).Among them, index 1
is the smallest, so the answer is 1.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **minRow()** which takes the two integers **n**, **m** as well as the **2D binary matrix a** as input parameters and returns the **minimum index** of the **row** which contains the **least number of 1's**.

Expected Time Complexity:O(n*m)

Expected Auxillary Space:O(1)

Constraints:

1 <= n,m <= 1000

0 <= a[i][j] <= 1

CODE

```
#User function Template for python3


class Solution:
    def minRow(self,n,m,a):
        ans=10**9
        index=0
        for i in range(n):
            count=0
            for j in range(m):
                if a[i][j]==1:
                    count+=1
            if ans>count:
                ans=count
                index=i
        if ans==0 and index==0:
            return 1
        return index+1
#code here
```