6342. First Completely Painted Row or Column

My Submissions (/contest/weekly-contest-343/problems/first-completely-painted-row-or-column/submissions/)
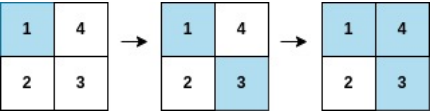
Back to Contest (/contest/weekly-contest-343/)

You are given a **0-indexed** integer array `arr`, and an `m x n` integer **matrix** `mat`. `arr` and `mat` both contain **all** the integers in the range $[1, m * n]$.

Go through each index `i` in `arr` starting from index `0` and paint the cell in `mat` containing the integer `arr[i]`.

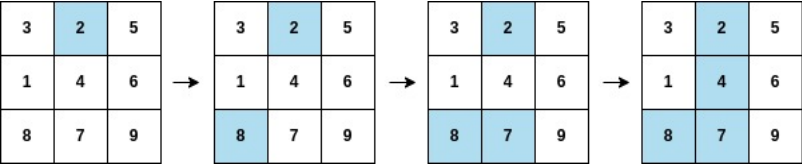Return *the smallest index* `i` *at which either a row or a column will be completely painted in* `mat`.

User Accepted:	0
User Tried:	3
Total Accepted:	0
Total Submissions:	3
Difficulty:	Medium

Example 1:


```
Input: arr = [1,3,4,2], mat = [[1,4],[2,3]]
Output: 2
Explanation: The moves are shown in order, and both the first row and second column of the matrix become fully painted at arr[2
```

Example 2:


```
Input: arr = [2,8,7,4,1,3,5,6,9], mat = [[3,2,5],[1,4,6],[8,7,9]]
Output: 3
Explanation: The second column becomes fully painted at arr[3].
```

Constraints:

- `m == mat.length`
- `n = mat[i].length`
- `arr.length == m * n`
- $1 <= m, n <= 10^5$
- $1 <= m * n <= 10^5$
- $1 <= arr[i], mat[r][c] <= m * n$
- All the integers of `arr` are **unique**.
- All the integers of `mat` are **unique**.

JavaScript ▾   ⟨⟩   ⟳   ⚙

```javascript
1    const initialize2DArray = (n, m) => [...Array(n)].map(() => Array(m).fill(0));
2
3 ▾  const firstCompleteIndex = (a, g) => {
4        let ma = new Map(), n = g.length, m = g[0].length;
5        for (let i = 0; i < n; i++)
6            for (let j = 0; j < m; j++)
7                ma.set(g[i][j], [i, j])
8        let row = initialize2DArray(n, 2), col = initialize2DArray(m, 2), t = 1;
9 ▾      for (const v of a) {
10           let [x, y] = ma.get(v);
11           row[x][0]++;
12           col[y][0]++;
13           if (row[x][0] == m && row[x][1] == 0) row[x][1] = t;
14           if (col[y][0] == n && col[y][1] == 0) col[y][1] = t;
15           t++;
16       }
```

```
17       row.sort((x, y) => x[1] - y[1]);
18       col.sort((x, y) => x[1] - y[1]);
19       return row[0][1] < col[0][1] ? row[0][1] - 1 : col[0][1] - 1;
20  };
```

☐ **Custom Testcase**   (Use Example Testcases)

▶ Run   ☁ Submit

Submission Result: Accepted (/submissions/detail/941893887/) ❓   (More Details ❯ (/submissions/detail/941893887/))

Share your acceptance!

Copyright © 2023 LeetCode

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

🇺🇸 United States (/region)