

## 2661. First Completely Painted Row or Column

Medium Topics Companies Hint

You are given a 0-indexed integer array `arr`, and an  $m \times n$  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`.

Example 1:

2

1	4		
2	3		

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]`.

eg. `mat`  $3 \times 4$

1	8	3	7
9	2	6	4
5	12	10	11

`arr` [ 2, 5, 7, 9, 1, 10, 11, 12, 4, 8, 3, 6 ]

$\rightarrow m = 3$   
 $n = 4$

1	(0.0)	7	(0.3)
2	(1.1)	8	(0.1)
3	(0.2)	9	(1.0)
4	(1.3)	10	(2.2)
5	(2.0)	11	(2.3)
6	(1.2)	12	(2.1)

$R =$  ~~0~~ ~~0~~ 0 0 ~~1~~ ~~1~~ 1 1 ~~2~~ 2 2 2  
 $C =$  ~~0~~ ~~0~~ ~~0~~ ~~1~~ 1 1 2 2 2 ~~3~~ 3 3

`arr` = [ ~~2~~ ~~5~~ ~~7~~ ~~9~~ ~~1~~ 10 11 12 4 8 3 6 ]

(1.1) (2.0) (0.3) (1.0) (0.0)

$\hookrightarrow$  col [0] full filled now  
res = 4

Accepted 1058 / 1058 testcases passed  
And... submitted at Jan 20, 2025 09:52

Editorial Solution

#### Runtime

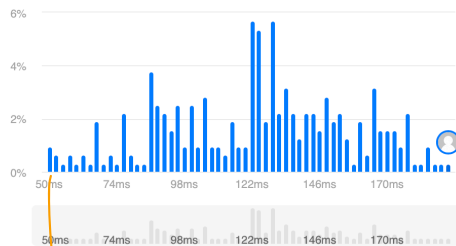
224 ms Beats 5.99%

Analyze Complexity

#### Memory

53.25 MB Beats 27.13%

Analyze Complexity



#### Code

Python3 Auto

```
1 class Solution:
2     def firstCompleteIndex(self, arr: List[int], mat: List[List[int]]) -> int:
3         m, n = len(mat), len(mat[0]) # row, col
4         dic = {}
5         for i in range(m):
6             for j in range(n):
7                 dic[mat[i][j]] = (i, j)
8
9         r = [0] * m
10        c = [0] * n
11        res = 0
12        for a in arr:
13            i, j = dic[a]
14            r[i] += 1
15            c[j] += 1
16
17            if r[i] == n or c[j] == m:
18                return res
19
20            res += 1
21
22        return res
23
```

Ln 10, Col 20 Saved



Python

```
class Solution:
    def firstCompleteIndex(self, arr: list[int], mat: list[list[int]]) -> int:
        m = len(mat)
        n = len(mat[0])
        # rows[i] := the number of painted grid in the i-th row
        rows = [0] * m
        # cols[j] := the number of painted grid in the j-th column
        cols = [0] * n
        # numToRow[num] := the i-th row of `num` in `mat`
        numToRow = [0] * (m * n + 1)
        # numToCol[num] := the j-th column of `num` in `mat`
        numToCol = [0] * (m * n + 1)

        for i, row in enumerate(mat):
            for j, num in enumerate(row):
                numToRow[num] = i
                numToCol[num] = j

        for i, a in enumerate(arr):
            rows[numToRow[a]] += 1
            if rows[numToRow[a]] == n:
                return i
            cols[numToCol[a]] += 1
            if cols[numToCol[a]] == m:
                return i
```

Almost the same thing.