

1428. Leftmost Column with at Least a One

Medium

👍 815

👎 101

🤍 Add to List

🔗 Share

(This problem is an **interactive problem**.)

A **row-sorted binary matrix** means that all elements are `0` or `1` and each row of the matrix is sorted in non-decreasing order.

Given a **row-sorted binary matrix** `binaryMatrix`, return the *index (0-indexed) of the leftmost column with a 1 in it*. If such an index does not exist, return `-1`.

You can't access the **Binary Matrix** directly. You may only access the matrix using a `BinaryMatrix` interface:

- `BinaryMatrix.get(row, col)` returns the element of the matrix at index `(row, col)` (0-indexed).
- `BinaryMatrix.dimensions()` returns the dimensions of the matrix as a list of 2 elements `[rows, cols]`, which means the matrix is `rows` x `cols`.

Submissions making more than `1000` calls to `BinaryMatrix.get` will be judged *Wrong Answer*. Also, any solutions that attempt to circumvent the judge will result in disqualification.

For custom testing purposes, the input will be the entire binary matrix `mat`. You will not have access to the binary matrix directly.

Example 1:

0	0
1	1

Input: mat = [[0,0],[1,1]]  
Output: 0

Example 2:

0	0
0	1

Input: mat = [[0,0],[0,1]]  
Output: 1

Example 3:

0	0
0	0

Input: mat = [[0,0],[0,0]]  
Output: -1

Example 4:

0	0	0	1
0	0	1	1
0	1	1	1

Input: mat = [[0,0,0,1],[0,0,1,1],[0,1,1,1]]  
Output: 1

Constraints:

- `rows == mat.length`
- `cols == mat[i].length`
- `1 <= rows, cols <= 100`
- `mat[i][j]` is either `0` or `1`.
- `mat[i]` is sorted in non-decreasing order.

Accepted 128,622

Submissions 248,851

Seen this question in a real interview before?

Yes

No

Companies

👤 i

^

0 ~ 6 months

6 months ~ 1 year

1 year ~ 2 years

Facebook

19

Related Topics

^

Array

Binary Search

Matrix

Interactive

Hide Hint 1

^

1. (Binary Search) For each row do a binary search to find the leftmost one on that row and update the answer.

Hide Hint 2

^

2. (Optimal Approach) Imagine there is a pointer `p(x, y)` starting from top right corner. `p` can only move left or down. If the value at `p` is `0`, move down. If the value at `p` is `1`, move left. Try to figure out the correctness and time complexity of this algorithm.

```
1 /**
2  * // This is the BinaryMatrix's API interface.
3  * // You should not implement it, or speculate about its implementation
4  * interface BinaryMatrix {
5  *     public int get(int row, int col) {}
6  *     public List<Integer> dimensions {}
7  * };
8  */
9
10 class Solution {
11     public int leftMostColumnWithOne(BinaryMatrix binaryMatrix) {
12     }
13 }
14
```

