Search in a 2D Matrix

We have to search in a two D matrix in O(log (mxn)) Complexity.

In the question they have given two properties

1. Each row is sorted in non-decreasing order
2. The first integer of each row is greater than last integer of previous row.

In this problem we have to do binary search on Rows then we have to find the row which has the value based on the above two properties.

```
ROWS, COLS = len(matrix), len(matrix[0])

top, bot = 0, ROWS-1

while top<=bot
    row = (top+bot)//2
    if target > matrix[row][-1]:
        top = row+1
    elif target < matrix[row][0]:
        bot = row-1
    else
        break
if not (top<=bot):
    return False
```

```
row = (top+ bot) // 2
l, r = 0, COLS - 1

while l<= r:
    m = (l+r) // 2
    if target > matrix[row][m]
        ~~m=~~ l = m+1
    elif target < matrix[row][m]
        r = m-1
    else
        return True
return False
```