Problem # 240: Search a 2D Matrix ii (Medium)

------------------------------------------------------------

<https://leetcode.com/problems/search-a-2d-matrix-ii/>

My Solution:

<https://leetcode.com/problems/search-a-2d-matrix-ii/discuss/1080169/Simple-Python-Solution-Runtime-beats-93.65>

Runtime beats 93.65%.

1. Let m be the number of rows in the matrix and n the number of columns.

2. Get the candidate rows for search.

Initialize row\_list to an empty list and this will hold the candidate rows.

Iterate through the rows one at a time. For each row find the first element and the last element and if either of them are equal to the target, return True. If target is between the first element and the last elements, then append the row number to the row\_list. If target is less than the first element, there is no point searching further. So, break out of the for loop.

3. Do the same for columns and get the candidate columns.

4. Search the matrix in candidate rows and columns only for element equal to target. If found return True. Otherwise at the end of the iteration, return False.

----------------------------------------------------------------------------------------------

class Solution:

def searchMatrix(self, matrix: List[List[int]], target: int) -> bool:

m = len(matrix)

n = len(matrix[0])

# get the candidate rows for search

row\_list = []

for i in range(m):

first\_ele = matrix[i][0]

last\_ele = matrix[i][n-1]

if target == first\_ele or target == last\_ele:

return True

if target > first\_ele and target < last\_ele:

row\_list.append(i)

elif target < first\_ele:

break

# get the candidate columns

col\_list = []

for j in range(n):

first\_ele = matrix[0][j]

last\_ele = matrix[m-1][j]

if target == first\_ele or target == last\_ele:

return True

if target > first\_ele and target < last\_ele:

col\_list.append(j)

elif target < first\_ele:

break

for row in row\_list:

for col in col\_list:

if target == matrix[row][col]:

return True

return False