

240. Search a 2D Matrix II

Medium Topics Companies

Write an efficient algorithm that searches for a value `target` in an `m x n` integer matrix `matrix`. This matrix has the following properties:

- Integers in each row are sorted in ascending from left to right.
- Integers in each column are sorted in ascending from top to bottom.

Example 1:

1	4	7	11	15
2	5	8	12	19
3	6	9	16	22
10	13	14	17	24
18	21	23	26	30

Input: `matrix = [[1,4,7,11,15],[2,5,8,12,19],[3,6,9,16,22],[10,13,14,17,24],[18,21,23,26,30]]`, `target = 5`
Output: `true`

Example 2:

1	4	7	11	15
2	5	8	12	19
3	6	9	16	22
10	13	14	17	24
18	21	23	26	30

Input: `matrix = [[1,4,7,11,15],[2,5,8,12,19],[3,6,9,16,22],[10,13,14,17,24],[18,21,23,26,30]]`, `target = 20`
Output: `false`

Constraints:

- `m == matrix.length`
- `n == matrix[i].length`
- `1 <= n, m <= 300`
- `-109 <= matrix[i][j] <= 109`
- All the integers in each row are **sorted** in ascending order.
- All the integers in each column are **sorted** in ascending order.

Brute force:
 $O(m \times n)$

	0	1	2	3	4
0	1	4	7	11	15
1	2	5	8	12	19
2	3	6	9	16	22
3	10	13	14	17	24
4	18	21	23	26	30

$$m = 5 \quad n = 5$$

start from Top right (or) bottom left

$$i = 0, j = n - 1$$

$$i = m - 1, j = 0$$

for Top right approach,

if $a_{ij} > \text{Target}$ then decrement column pointer j .

else if $a_{ij} < \text{Target}$ then increment row pointer i .

$$i = 0 \quad j = n - 1$$

while ($i < m$ && $j \geq 0$)

{

if ($a_{ij} == \text{Target}$)

return true

else if ($a_{ij} > \text{Target}$)

$j--$

else

$i++$

}

return false

$$T(n) : O(m+n)$$

$$S(n) : O(1)$$

Q: why can't we start from Top left (or) bottom right?

here we don't have two paths to take bcoz both are increasing paths or both are decreasing paths - So we can't make a decision.

	0	1	2	3	4
0	1	4	7	11	15
1	2	5	8	12	19
2	3	6	9	16	22
3	10	13	14	17	24
4	18	21	23	26	30

here we have two paths to take for two conditions.

i.e. $a_{ij} > T$ then move \leftarrow

$a_{ij} < T$ then move \downarrow