240. Search a 2D Matrix II



Write an efficient algorithm that searches for a value target in an $m \times n$ integer matrix target. 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:

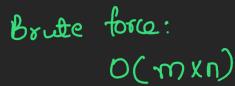
1	4	7	11	15	
2	5	8	12	19	
3	6	9	16	22	
10	13	14	17	24	
18	21	23	26	30	

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
- $-10^9 \le \text{matrix[i][j]} \le 10^9$
- 20 1 1110 21 27 [2] [3] 1 20
- All the integers in each row are **sorted** in ascending order.
- All the integers in each column are **sorted** in ascending order.



	D	ſ	2	3	. φ
6	1	Ч	7	11	15
1	2	5	8	12	19
2	3	6	9	16	22
3	10	13	ſΨ	(7	24
4	18	21	23	26	30

return false

[(n) : D (m+n) S(n) : D (1)

m=5 n=5

9: why can't we start from Top (eft (or) bottom right?

rve	D	1	2	3	. 4
6		Ч	7	11	(15
1	2	5	8	12	19
2	3	6	9	16	22
3	10	13	ſΨ	17	24
Ч	(18)	21	23	26	30
	61234	2 2 3 10 4	0 1 4 1 2 5 2 3 6 3 10 13	0 1 4 7 1 2 5 8 2 3 6 9 3 10 13 14 4 18 21 23	1 2 3 8 12 2 3 6 9 16 3 10 13 14 17 18 21 23 26 18

here we have two
Pouths to take for two
conditions.
i.e. a; > T then move f
a; < T then move f