240. Search a 2D Matrix II

Medium 8382 140 Add to List Share

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`
- `-10^9 <= matrix[i][j] <= 10^9`
- All the integers in each row are **sorted** in ascending order.
- All the integers in each column are **sorted** in ascending order.
- `-10^9 <= target <= 10^9`

Accepted 658,032 Submissions 1,327,735

Seen this question in a real interview before? Yes No

```cpp
class Solution
{
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int n = matrix.size(), i = 0, j = matrix[0].size()-1;
        while(i < n && ~j){
            int cell = matrix[i][j];
 if(cell == target) return true;
 else if(cell > target) j--;
 else i++;
        }
        return false;
    }
};
```

Te... Run Cod... De...

Accepted Runtime: 6 ms

Your input [[1,4, [2,5,8

Output Diff

Expected true