

搜索二维矩阵 II

编写一个高效的算法来搜索 $m \times n$ 矩阵中的一个目标值。

该矩阵具有以下特性：

每行的元素从左到右升序排列。

每列的元素从上到下升序排列。

例如，

考虑下面的矩阵：

[[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]]

给定目标值 $target = 5$, 返回 `true`。

给定目标值 $target = 20$, 返回 `false`。

<https://leetcode-cn.com/problems/search-in-rotated-sorted-array-ii/description/>

```
public boolean searchMatrix(int[][] matrix, int target) {  
    if (matrix == null || matrix.length == 0) {  
        return false;  
    }  
  
    if (matrix[0] == null || matrix[0].length == 0) {  
        return false;  
    }  
  
    int rowBegin = 0;  
    int rowEnd = matrix.length - 1;  
    int colBegin = 0;  
    int colEnd = matrix[0].length - 1;  
  
    while(colEnd >= colBegin && rowBegin <= rowEnd) {  
        int temp = matrix[rowBegin][colEnd];  
        if(target == temp) {  
            return true;  
        }  
        else if (target < temp) {  
            colEnd--;  
        }  
        else if (target > temp) {  
            rowBegin++;  
        }  
    }  
  
    return false;  
}
```