/\*\*

\* @param {number[][]} matrix

\* @param {number} k

\* @return {number}

\*/

var kthSmallest = function(matrix, k) {

const n = matrix.length;

let left = matrix[0][0];

let right = matrix[n-1][n-1];

// 二分查找

while (left < right) {

const mid = left + Math.floor((right - left) / 2);

const count = countLessOrEqual(matrix, mid);

if (count < k) {

left = mid + 1;

} else {

right = mid;

}

}

return left;

};

// 统计矩阵中小于等于target的元素数量

function countLessOrEqual(matrix, target) {

const n = matrix.length;

let count = 0;

let row = ️0;

let col = n - 1;

// 从右上角开始搜索

while (row < n && col >= 0) {

if (matrix[row][col] <= target) {

count += col + 1; // 当前行有col+1个元素<=target

row++; // 移动到下一行

} else {

col--; // 向左移动一列

}

}

return count;

}