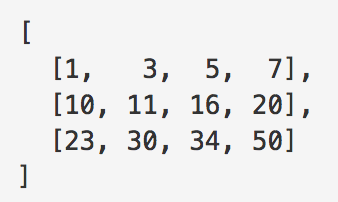
74. Search a 2D Matrix

要求：编写一个搜索m×n矩阵中的值的有效算法。 该矩阵具有以下属性：

每行整数从左到右由小到大排序。

每行的第一个整数大于上一行的最后一个整数。

比如：



给出target=3，返回true。

这道题标识为middle。

这道题使用了二分查找，最主要的部分是将二维数组怎样转换成一维数组。

设置二维数组的第一个元素是left=0，最后一个元素是right=row\*col-1。

然后二分查找需要求出数组的中间位置：mid=parseInt((left+right)/2，将这个中间位置转化为对应的二维数组的中间位置：matrix[parseInt(mid/col)][mid%col]。

/\*\*

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

\* @param {number} target

\* @return {boolean}

\*/

var searchMatrix = function(matrix, target) {

let row=matrix.length;

if(row===0){

return false;

}

let col=matrix[0].length;

let left=0;

let right=row\*col-1;

while(left<=right){

let mid=parseInt((left+right)/2);

if(matrix[parseInt(mid/col)][mid%col]===target){

return true;

}else if(matrix[parseInt(mid/col)][mid%col]<target){

left=mid+1;

}else{

right=mid-1;

}

}

return false;

};