

编程题_README

1.题目简介

编写一个算法实现搜索 $M \times N$ 矩阵 `matrix` 中的一个目标值 `target`。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。

要求：时间复杂度小于 $O(mn)$ 。

2.我的思路

(1)由于要求的时间复杂度小于 $O(mn)$ ，所以我们不能采用传统暴力的遍历算法来解决。我们发现二维矩阵的每一行满足从大到小的规律排列，是满足二分搜索条件的，故而采取对每一行进行二分搜索的办法来进行查找。对于二维矩阵的每一行，如果满足 `Matrix[row][0] <= target && Matrix[row][N - 1] >= target`，即说明要查找的目标值 `target` 的大小位于每一行最小的值和最大的值中间，这个时候我们就可以对每一行二分查找通过二分查找的方式进行搜索是否存在 `target` 这个元素。

由于矩阵的行数为 M ，矩阵的列数为 N ，对于每一行进行二分搜索操作，故而该算法的时间复杂度为 $O(m \log n)$ ，满足题目的要求。

(2)有没有时间复杂度为 $O(m+n)$ 的方法呢？对于二维矩阵每一行来说是从左到右升序排列，每一列来说从上到下升序排列，我们可以发现左上角的那一个元素是该行最大的，该列最小的，所以从左上角的那个元素开始搜索：如果 `target` 小于左上角的这个元素，说明 `target` 不会出现在该元素所在列下面的元素中，故而 `N--`，将此列省去；如果 `target` 大于左上角的这个元素，说明 `target` 不会出现在该元素所在行左边的元素中，故而 `row++`，将此行省去；如果 `target` 等于左上角的这个元素，说明找到该元素，返回 `true`，如果在搜索后还没有找到，返回 `false`。

在搜索的过程中，如果没有找到元素 `target`，那么 `row` 的值最多从 0 增加到 $M-1$ ，`N` 的值最多从 N 减少到 1，总的搜索次数为 $M+N$ ，时间复杂度为 $O(m+n)$ ，满足题目的要求。

3.主要函数说明

(1)`SearchTarget_BS(vector<vector<int>>matrix, int target, int M, int N)`函数

这个函数的输入为二维矩阵 `matrix`，目标值 `target`，行数 `M`，列数 `N`，作用是使用时间复杂度为 $O(m \log n)$ 的二分查找进行搜索。这里注意需要对每一行进行检查，检查每一行是否满足 `Matrix[row][0] <= target && Matrix[row][N - 1] >= target`，即说明要查找的目标值 `target` 的大小位于每一行最小的值和最大的值中间。之后便可以进行二分搜索来查找该行是否存在目标值 `target`。

(2) `SearchTarget_Z(vector<vector<int>>matrix, int target, int M, int N)` 函数

这个函数的输入为二维矩阵 `matrix`，目标值 `target`，行数 `M`，列数 `N`，作用是使用时间复杂度为 $O(m+n)$ 的 Z 字形查找进行搜索。从左上角的那个元素开始，通过不断比较左上角元素和 `target` 的大小，来决定要将左上角元素所属的行还是列省去，最后得出结果。

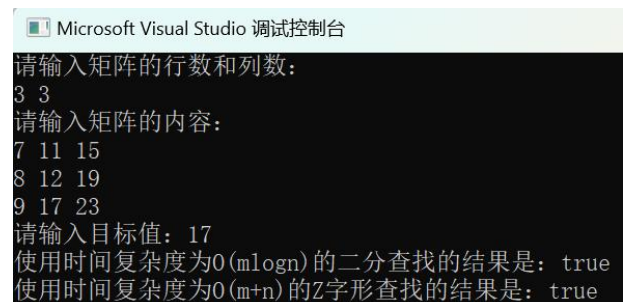
4. 三个测试用例

用例 1:

输入: `matrix:[[7,11,15],[8,12,19],[9,17,23]]`

`target:17`

输出: `true`



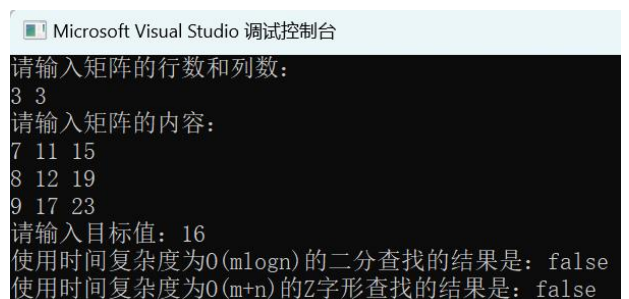
```
Microsoft Visual Studio 调试控制台
请输入矩阵的行数和列数:
3 3
请输入矩阵的内容:
7 11 15
8 12 19
9 17 23
请输入目标值: 17
使用时间复杂度为O(mlogn)的二分查找的结果是: true
使用时间复杂度为O(m+n)的Z字形查找的结果是: true
```

用例 2:

输入: `matrix:[[7,11,15],[8,12,19],[9,17,23]]`

`target:16`

输出: `false`



```
Microsoft Visual Studio 调试控制台
请输入矩阵的行数和列数:
3 3
请输入矩阵的内容:
7 11 15
8 12 19
9 17 23
请输入目标值: 16
使用时间复杂度为O(mlogn)的二分查找的结果是: false
使用时间复杂度为O(m+n)的Z字形查找的结果是: false
```

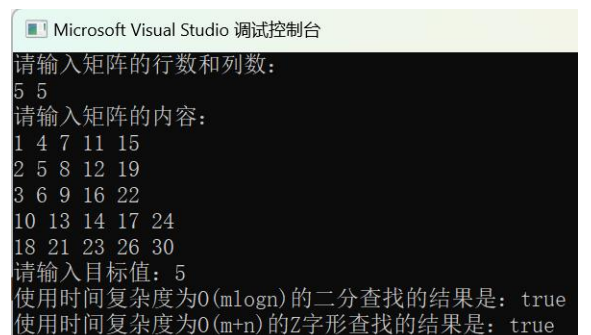
用例 3:

输入:

`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` 输出: `true`



```
Microsoft Visual Studio 调试控制台
请输入矩阵的行数和列数:
5 5
请输入矩阵的内容:
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
请输入目标值: 5
使用时间复杂度为O(mlogn)的二分查找的结果是: true
使用时间复杂度为O(m+n)的Z字形查找的结果是: true
```

5. 附：有没有时间复杂度为 $O(m+n)$ 的方法？

见上文和 `cpp` 文件中 `SearchTarget_Z` 函数的方法。