

A2_README

一、算法简介

本算法是基于二维矩阵的搜索问题，通过有效的搜索方式确定目标值是否存在于矩阵中。算法从矩阵左下角开始搜索，利用有序性，逐步缩小搜索范围，以提高搜索效率。

二、实现思路

1. 搜索起点的选择：由矩阵的有序性选择从矩阵的左下角作为起点开始搜索，因为该位置元素是当前行的最小值，当前列的最大值，这样在不断移动过程中可以根据比较结果缩小搜索范围。
2. 搜索过程：从左下角开始，通过与目标值比较来决定向右移动还是向上移动：如果当前元素等于目标值，则返回 `true`。如果当前元素大于目标值，则向上移动一行。如果当前元素小于目标值，则向右移动一列。重复上述步骤直到搜索结束或者找到目标值。
3. 搜索结束条件：搜索超出矩阵边界或找到目标值。
4. 时间复杂度分析：每次移动都能排除一行或一列，因此算法的时间复杂度为 $O(m + n)$ ，其中 m 是矩阵的行数， n 是矩阵的列数。

三、测试用例

1.

```
Please enter m: 3
Please enter n: 3
Please enter matrix:
7 11 15
8 12 19
9 17 23
Please enter target: 17
true
```

2.

```
Please enter m: 3
Please enter n: 3
Please enter matrix:
7 11 15
8 12 19
9 17 23
Please enter target: 16
false
```

3.

```
Please enter m: 4
Please enter n: 5
Please enter matrix:
1 4 7 11 15
2 5 8 12 19
3 6 9 16 22
10 13 14 17 24
Please enter target: 19
true
```