查询二维矩阵L74.note矩阵按照箭头方向递增。

74. Search a 2D Matrix

Description | Hints | Submissions | Discuss | Solution

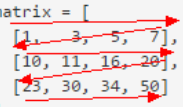
Pick One

Write an efficient algorithm that searches for a value in an *m* x *n* matrix. This matrix has the following properties:

- Integers in each row are sorted from left to right.
- The first integer of each row is greater than the last integer of the previous row.

Example 1:

```
Input:
matrix = [
  [1,   3,  5,  7],
  [10, 11, 16, 20],
  [23, 30, 34, 50]
]
target = 3
Output: true
```



Example 2:

```
Input:
matrix = [
  [1,   3,  5,  7],
  [10, 11, 16, 20],
  [23, 30, 34, 50]
]
target = 13
Output: false
```

```java
public class L74 {
    /*
     * 这道题是利用二分法
     */
     public boolean searchMatrix(int[][] matrix, int target) {
             if(matrix == null || matrix.length == 0 || matrix[0].length == 0) {
                 return false;
             }

             int m = matrix.length;
             int n = matrix[0].length;

             int start = 0;
             int end = m * n - 1;

             while (start <= end) {
                int mid = (start + end) / 2;
                int midX = mid / n;
                int midY = mid % n;

                if(matrix[midX][midY] == target)
                    return true;

                if(matrix[midX][midY] < target)
                    start = mid + 1;
                else {
                    end = mid - 1;
                }
            }
             return false;
    }
}
```