```java
public boolean searchMatrix(int[][] matrix, int target) {
    if (matrix == null || matrix.length == 0 || matrix[0].length == 0) {
        return false;
    }
    int row = matrix.length;
    int col = matrix[0].length;
    int start = 0;
    int end = row * col - 1;
    while (start + 1 < end) {
        int mid = start + (end - start) / 2;
        int x = mid / col ;
        int y = mid % col ;
        if (matrix[x][y] == target) {
            end = mid;
        }
        else if (matrix[x][y] < target) {
            start = mid;
        }
        else {
            end = mid;
        }
    }
    if (matrix[start / col][start % col] == target) {
        return true;
    }
    if (matrix[end / col][end % col] == target) {
        return true;
    }

    return false;
}
```

# 搜索二维矩阵 II

编写一个高效的算法来搜索 m x n 矩阵中的一个目标值。
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。
例如，
考虑下面的矩阵：
[ [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] ]
给定目标值 target = 5, 返回 true。
给定目标值 target = 20, 返回 false。
https://leetcode-cn.com/problems/search-in-rotated-sorted-array-ii/description/