一、题目说明

题目240. Search a 2D Matrix II，从一个m*n的二维矩阵查找一个整数，每一行从左到右递增，每一列从上到下递增。

二、我的解答

先计算矩阵中点`matrix[row_mid][col_mid]`，然后将矩阵分成4个区间：

```
class Solution{
    public:
        bool dfs(vector<vector<int> >& matrix,int target,int start_x, int end_x,
int start_y, int end_y){
            if(start_x > end_x || start_y > end_y)
                return false;
            if(start_x == end_x && start_y == end_y)
                return matrix[start_x][start_y] == target;
            if(target < matrix[start_x][start_y] || target > matrix[end_x]
[end_y])
                return false;
            int mid_x = start_x + (end_x - start_x) / 2;
            int mid_y = start_y + (end_y - start_y) / 2;
            if(matrix[mid_x][mid_y] == target) {
                return true;
            } else if(matrix[mid_x][mid_y] > target) {
                //在1,2,3区域找
                return dfs(matrix, target, start_x, mid_x - 1, start_y, mid_y -
1)
                    || dfs(matrix, target, mid_x, end_x, start_y, mid_y - 1)
                    || dfs(matrix, target, start_x, mid_x - 1, mid_y, end_y);
            } else {
                //在2,3,4区域找
                return dfs(matrix, target, mid_x + 1, end_x, mid_y + 1, end_y)
                    || dfs(matrix, target, start_x, mid_x, mid_y + 1, end_y)
                    || dfs(matrix, target, start_x + 1, end_x, start_y, mid_y);
            }
        }

        bool searchMatrix(vector<vector<int> >& matrix,int target){
            if(matrix.size()<1){
                return false;
            }
            row = matrix.size();
            col = matrix[0].size();
            bool result = dfs(matrix,target,0,row-1,0,col-1);
            if(result) return true;
            else return false;
        }
    private:
        int row,col;
};
```

性能如下：

```
Runtime: 148 ms, faster than 24.09% of C++ online submissions for Search a 2D
Matrix II.
Memory Usage: 13.1 MB, less than 6.67% of C++ online submissions for Search a 2D
Matrix II.
```

三、优化措施

从左下角（右上角也行）出发，比较绝妙的解答：

```
class Solution{
    public:
        //从左下角（或者 右上角）出发，判断
        bool searchMatrix(vector<vector<int> >& matrix,int target){
            if(matrix.size()<1){
                return false;
            }
            row = matrix.size();
            col = matrix[0].size();
            int curRow = row-1;
            int curCol = 0;
            while(curRow<row && curCol<col){
                if(matrix[curRow][curCol] == target) return true;
                else if(matrix[curRow][curCol] < target){
                    curCol ++;
                }else if(matrix[curRow][curCol] > target){
                    curRow--;
                }
            }
            return false;
        }
    private:
        int row,col;
};
```

性能如下：

```
Runtime: 72 ms, faster than 61.20% of C++ online submissions for Search a 2D
Matrix II.
Memory Usage: 13 MB, less than 55.56% of C++ online submissions for Search a 2D
Matrix II.
```