74. Search a 2D Matrix

link

这里写了两次二分查找, 虽然麻烦了, 但是找小于之前值的二分查找, 条件必须是while(start<end), 最后arr[mid+1]>target一定不能带等号, 这里错了好几次了, 其实当作一位数组写就行了.

普通二分查找, high最后返回的一定是小一点的值, low是大一点的.

```
    public boolean searchMatrix(int[][] matrix, int target) {
        if(matrix.length == 0 || matrix[0].length == 0) return false;
        int i = matrix.length, j = matrix[0].length;
        if(matrix[0][0] > target || matrix[i - 1][j - 1] < target)
            return false;
        int start = 0, end = i - 1;
//条件
        while(start < end){
            int mid = start + (end - start) / 2;
            if(matrix[mid][0] == target)
                return true;
            else if(matrix[mid][0] > target){
                if(mid > 0 && matrix[mid-1][0] <= target){
                    start = mid - 1;
                    break;
                }
                end = mid - 1;
            }else{
                //这里大于不能加等于, 错了好几次.
                if(mid < i - 1 && matrix[mid + 1][0] > target){
                    start = mid;
                    break;
                }
                start = mid + 1;
            }
        }
        i = start;
        start = 0;
        end = j - 1;
        while(start <= end){
            int mid = start + (end - start) / 2;
            if(matrix[i][mid] == target)
                return true;
            else if(matrix[i][mid] > target){
                end = mid - 1;
            }else{
                start = mid + 1;
            }
        }
        return false;
```

```
}
```

一位数组, 算一下index就行了(判断mn部越界就好了)

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