240. Search a 2D Matrix II

```
class Solution {
    public boolean searchMatrix(int[][] matrix, int target) {
        if(matrix.length == 0){
            return false;
        }
        int first = 0;
        int second =matrix[0].length - 1;
        while(inbound(first, second, matrix)){
            if(matrix[first][second] == target){
                return true;
            }
            else if(matrix[first][second] > target){
                second--;
            } else {
                first++;
            }
        }
        return false;
    }

    private boolean inbound(int first, int second, int[][] matrix){
        return first >= 0 && second >= 0 && first < matrix.length && second < matrix[0].length;
    }
}
```

148. Sort List

题目：将链表排序

Hint：Divide and conquer

```
class Solution {
    public ListNode sortList(ListNode head) {
        if(head == null || head.next == null){
            return head;
        }
        ListNode slow = head;
        ListNode fast = head.next;
        while(fast != null && fast.next != null){
            slow = slow.next;
            fast = fast.next.next;
        }
        ListNode left = head;
        ListNode right = slow.next;
        slow.next = null;
        left = sortList(left);
        right = sortList(right);

        ListNode start = new ListNode(0);
        ListNode node = start;
        while(left != null || right != null){
            if(left == null){
                node.next = right;
                break;
            }
            if(right == null){
                node.next = left;
                break;
            }
            if(left.val < right.val){
                node.next = left;
                node = node.next;
                left = left.next;
            } else{
                node.next = right;
                node = node.next;
                right = right.next;
            }
        }
        return start.next;
    }
}
```