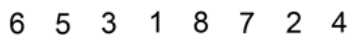
147 对链表进行插入排序

Label：链表

对链表进行插入排序



- 迭代后反转

```
class Solution {
    public ListNode insertionSortList(ListNode head) {

        ListNode newHead = new ListNode(Integer.MAX_VALUE);
        ListNode curr = head;
        while (curr != null) {
           // 遍历新链表寻找位置
           ListNode newCurr = newHead;
           while (newCurr.next != null && newCurr.next.val > curr.val) {
               newCurr = newCurr.next;
           }
           // 插入结点
           ListNode temp = newCurr.next;
           newCurr.next = new ListNode(curr.val);
           newCurr.next.next = temp; // 链接

           curr = curr.next;
        }

        // 反转链表
        return reverseList(newHead.next);

    }

    private ListNode reverseList(ListNode head) {
        if(head == null || head.next == null) {
            return head;
        }
        ListNode  reverseHead = reverseList(head.next);  // 要一直返回，这个值从最底层开始就是不变的
        head.next.next = head;
        head.next = null;  // 防止 顶层 出现 cycle
        return reverseHead;
    }
}
```

- 迭代，不反转

```
class Solution {
    public ListNode insertionSortList(ListNode head) {

        if(head == null) return head;

        // 链表初始化操作
        ListNode dummyHead = new ListNode(0); // 引入哑结点，用来顺序记录，方便返回
        dummyHead.next = head;
        ListNode lastSorted = head;            // 链表已经排好序的最后一个结点
        ListNode curr = head.next;             // 待插入结点

        // 插入排序
        while (curr != null) {
            if (lastSorted.val <= curr.val) {     // 比最大的节点还大，lastSorted后移一位
                lastSorted = lastSorted.next;
            } else {                              // 从排序好的部分开始向后遍历
                ListNode prev = dummyHead;       // 从链表头开始遍历 prev是准备插入位置的前一个节点
                while(prev.next.val <= curr.val){ // 从小到大，从左到右
                    prev = prev.next;
                }
                // 插入 curr
                lastSorted.next = curr.next;
                curr.next = prev.next; // 链接后面
                prev.next = curr;
            }
            curr = lastSorted.next;
        }
        // 返回排好序的链表
        return dummyHead.next;
    }
}
```