147. Insertion Sort List

Description | Hints | Submissions | Discuss | Solution

Pick One

Sort a linked list using insertion sort.

A graphical example of insertion sort. The partial sorted list (black) initially contains only the first element in the list.

With each iteration one element (red) is removed from the input data and inserted in-place into the sorted list

Algorithm of Insertion Sort:

1. Insertion sort iterates, consuming one input element each repetition, and growing a sorted output list.
2. At each iteration, insertion sort removes one element from the input data, finds the location it belongs within the sorted list, and inserts it there.
3. It repeats until no input elements remain.

Example 1:

```
Input: 4->2->1->3
Output: 1->2->3->4
```

Example 2:

```
Input: -1->5->3->4->0
Output: -1->0->3->4->5
```

```java
/*
 * 这道题目和Sort List类似，要求在链表上实现一种排序算法，这道题是指定实现插入排序算法。插入排序是一种O(n^2)复杂度的算法
 * 基本思想就不谈了，就是每次循环找到一个元素在当前排好的结果中相对应的位置，然后插进去，经过n次迭代之后就得到排好序的算法。了解了
 * 思路之后就是链表的基本操作了，搜索并进行相应的插入。时间复杂度是一样的，空间复杂度为O(1).
 */
public class L147 {
    public class ListNode {
              int val;
              ListNode next;
              ListNode(int x) { val = x; }
    }

     public ListNode insertionSortList(ListNode head) {
         if (head == null) {
             return null;
         }
         //这里是用一个辅助指针来做表头避免处理改变head的时候边界情况。
         ListNode helper = new ListNode(0);
         ListNode pre = helper;
         ListNode cur = head;

         while (cur != null) {
             ListNode next = cur.next;
             pre = helper;
             //找到插入的位置，没有小的就排在最后
             while (pre.next != null && pre.next.val <= cur.val) {
                 pre = pre.next;
             }
             //下面三个语句是改变指针，将cur插入进来
             cur.next = pre.next;
             pre.next = cur;
             cur = next;
         }
         //这里返回的是helper而不是head
         return helper.next;
     }
}
```