Insertion Sort List

Sort a linked list using insertion sort.

Solution 1

```
public ListNode insertionSortList(ListNode head) {
        if( head == null ){
            return head;
        }

        ListNode helper = new ListNode(0); //new starter of the sorted list
        ListNode cur = head; //the node will be inserted
        ListNode pre = helper; //insert node between pre and pre.next
        ListNode next = null; //the next node will be inserted
        //not the end of input list
        while( cur != null ){
            next = cur.next;
            //find the right place to insert
            while( pre.next != null && pre.next.val < cur.val ){
                pre = pre.next;
            }
            //insert between pre and pre.next
            cur.next = pre.next;
            pre.next = cur;
            pre = helper;
            cur = next;
        }

        return helper.next;
    }
```

written by sbvictory original link here

Solution 2

One of the quotes is

> For God's sake, don't try sorting a linked list during the interview

http://steve-yegge.blogspot.nl/2008/03/get-that-job-at-google.html

So it might be better to actually copy the values into an array and sort them there.

written by orekhov.volodya original link here

Solution 3

```
public class Solution {
public ListNode insertionSortList(ListNode head) {
    ListNode helper=new ListNode(0);
    ListNode pre=helper;
    ListNode current=head;
    while(current!=null) {
        pre=helper;
        while(pre.next!=null&&pre.next.val<current.val) {
            pre=pre.next;
        }
        ListNode next=current.next;
        current.next=pre.next;
        pre.next=current;
        current=next;
    }
    return helper.next;
}
```

}

written by isly original link here

From Leetcoder.