Merge Two Sorted Lists

Merge two sorted linked lists and return it as a new list. The new list should be made by splicing together the nodes of the first two lists.

Solution 1

```
class Solution {
public:
    ListNode *mergeTwoLists(ListNode *l1, ListNode *l2) {
        if(l1 == NULL) return l2;
        if(l2 == NULL) return l1;

        if(l1->val < l2->val) {
            l1->next = mergeTwoLists(l1->next, l2);
            return l1;
        } else {
            l2->next = mergeTwoLists(l2->next, l1);
            return l2;
        }
    }
};
```

This solution is not a tail-recursive, the stack will overflow while the list is too long :)
http://en.wikipedia.org/wiki/Tail_call

written by GZShi original link here

Solution 2

```
class Solution {
public:
    ListNode *mergeTwoLists(ListNode *l1, ListNode *l2) {
        ListNode dummy(INT_MIN);
        ListNode *tail = &dummy;

        while (l1 && l2) {
            if (l1->val < l2->val) {
                tail->next = l1;
                l1 = l1->next;
            } else {
                tail->next = l2;
                l2 = l2->next;
            }
            tail = tail->next;
        }

        tail->next = l1 ? l1 : l2;
        return dummy.next;
    }
};
```

written by xiaohui7 original link here

Solution 3

Hello every one, here is my code, simple but works well:

```
public class Solution {
    public ListNode mergeTwoLists(ListNode l1, ListNode l2) {
        if(l1 == null){
            return l2;
        }
        if(l2 == null){
            return l1;
        }

        ListNode mergeHead;
        if(l1.val < l2.val){
            mergeHead = l1;
            mergeHead.next = mergeTwoLists(l1.next, l2);
        }
        else{
            mergeHead = l2;
            mergeHead.next = mergeTwoLists(l1, l2.next);
        }
        return mergeHead;
    }
}
```

written by zwangbo original link here