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.

Java解决方法：

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(l1, l2.next);

return l2;

}

}

C语言解决方式：

struct ListNode\* newNode(int data)

{

struct ListNode \*n = malloc(sizeof(struct ListNode));

n->val = data;

n->next = NULL;

return n;

}

struct ListNode\* mergeTwoLists(struct ListNode\* l1, struct ListNode\* l2)

{

if(l1 == NULL && l2 == NULL)

return;

if(l1 == NULL)

return l2;

else if(l2 == NULL)

return l1;

struct ListNode \*aCurrent = l1;

struct ListNode \*bCurrent = l2;

struct ListNode \*res = NULL;

struct ListNode \*temp = NULL;

while(aCurrent != NULL && bCurrent != NULL)

{

if(aCurrent->val <= bCurrent->val) {

struct ListNode \*n = newNode(aCurrent->val);

if(res == NULL) {

res = n;

temp = res;

}else{

temp->next = n;

temp = temp->next;

}

aCurrent = aCurrent->next;

}else{

struct ListNode \*n = newNode(bCurrent->val);

if(res == NULL) {

res = n;

temp = res;

}else{

temp->next = n;

temp = temp->next;

}

bCurrent = bCurrent->next;

}

}

while(aCurrent != NULL)

{

struct ListNode \*n = newNode(aCurrent->val);

temp->next = n;

temp = temp->next;

aCurrent = aCurrent->next;

}

while(bCurrent != NULL)

{

struct ListNode \*n = newNode(bCurrent->val);

temp->next = n;

temp = temp->next;

bCurrent = bCurrent->next;

}

return res;

}