## 题目描述：

## 输入两个单调递增的链表，输出两个链表合成后的链表，当然我们需要合成后的链表满足单调不减规则。

## 解题思路：

1. 递归用法
2. 非递归法

## 算法描述：

1. 递归法（简洁） ☆☆☆☆☆

/\*

struct ListNode {

int val;

struct ListNode \*next;

ListNode(int x) :

val(x), next(NULL) {

}

};\*/

class Solution {

public:

ListNode\* Merge(ListNode\* pHead1, ListNode\* pHead2)

{

if (pHead1 == NULL)

return pHead2;

if (pHead2 == NULL)

return pHead1;

while (pHead1 != NULL && pHead2 != NULL)

{

if (pHead1->val <= pHead2->val )

{

pHead1->next = Merge(pHead1->next,pHead2);

return pHead1;

}

else

{

pHead2->next = Merge(pHead1,pHead2->next);

return pHead2;

}

}

}

};

1. 非递归：

class Solution {

public:

ListNode\* Merge(ListNode\* pHead1, ListNode\* pHead2)

{

ListNode\* p = NULL;

ListNode\* Head = NULL;

if (pHead1 == NULL)

return pHead2;

if (pHead2 == NULL)

return pHead1;

if (pHead1->val <= pHead2->val)

{

Head = pHead1;

pHead1= pHead1->next;

}

else

{

Head = pHead2;

pHead2 = pHead2->next;

}

p = Head;

while (pHead1 != NULL && pHead2 != NULL)

{

if (pHead1->val <= pHead2->val )

{

p->next = pHead1;

pHead1 = pHead1->next;

p = p->next;

}

else

{

p->next = pHead2;

pHead2 = pHead2->next;

p = p->next;

}

}

if (pHead1 == NULL)

p->next = pHead2;

if (pHead2 == NULL)

p->next = pHead1;

return Head;

}

};