思路不清晰下的没有头，返回错误：

class Solution {

public:

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

ListNode \*p,\*q,\*s;

p=l1;

q=l2;

if(p==NULL&&q!=NULL) return l2;

if(p!=NULL&&q==NULL) return l1;

while(p!=NULL&&q!=NULL)

{

if(p->val>q->val)

{

s=q->next;

q->next=p;

p=p->next;

q=s;

}

else

{

s=p->next;

p->next=q;

q=q->next;

p=s;

}

}

if(p==NULL&&q!=NULL&&(l2->val<l1->val))

{

return l2;

}

else if(p==NULL&&q!=NULL&&(l2->val>=l1->val))

{

p->next=s;

return l1;

}

if(p!=NULL&&q==NULL&&(l2->val<l1->val))

{

q->next=s;

return l2;

}

else if(p!=NULL&&q==NULL&&(l2->val>=l1->val))

{

return l1;

}

}

};

不说了，直接上代码：

class Solution {

public:

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

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

return l1;

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

return l2;

ListNode \*l3;

ListNode \*p1,\*p2,\*p3;

p1=l1; //遍历l1

p2=l2; //遍历l2

if(p1->val<=p2->val) //确定l3的头

{

l3=l1;

p1=p1->next;

}

else

{

l3=l2;

p2=p2->next;

}

p3=l3; //p3始终指向l3的尾结点

while(p1!=NULL&&p2!=NULL)

{

if(p1->val<=p2->val)

{

p3->next=p1;

p1=p1->next;

p3=p3->next;

}

else

{

p3->next=p2;

p2=p2->next;

p3=p3->next;

}

}

if(p1==NULL)

{

p3->next=p2;

}

if(p2==NULL)

{

p3->next=p1;

}

return l3;

}

};