

合并两个有序链表

将两个升序链表合并为一个新的升序链表并返回。新链表是通过拼接给定的两个链表的所有节点组成的。

示例：

输入：1->2->4, 1->3->4

输出：1->1->2->3->4->4

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
class Solution {
public:
    ListNode* mergeTwoLists(ListNode* l1, ListNode* l2) {
        if(l1==nullptr)
            return l2;
        if(l2==nullptr)
            return l1;
        if(l1->val<=l2->val)
        {
            l1->next=mergeTwoLists(l1->next,l2);
            return l1;
        }
        else
        {
            l2->next=mergeTwoLists(l1,l2->next);
            return l2;
        }
    }
};
/**
class Solution {
public:
    ListNode* mergeTwoLists(ListNode* l1, ListNode* l2) {
        ListNode* preHead = new ListNode(-1);
```

```
ListNode* prev = preHead;
while (l1 != nullptr && l2 != nullptr) {
    if (l1->val < l2->val) {
        prev->next = l1;
        l1 = l1->next;
    } else {
        prev->next = l2;
        l2 = l2->next;
    }
    prev = prev->next;
}

// 合并后 l1 和 l2 最多只有一个还未被合并完，我们直接将链表末尾指向未合并
// 完的链表即可
prev->next = l1 == nullptr ? l2 : l1;

return preHead->next;
}
};

*/
```