1. 题目
2. 题干

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。

1. 示例

**输入：**1->2->4, 1->3->4

**输出：**1->1->2->3->4->4

1. 题解
2. 思路

这是一道简单的题目，通过循环比较两个链表的每个值，并创建一个新的链表用于存储合并后的值。注意，创建新的链表要创建两个指针，一个指针指向头结点是不动的，用于方便最后返回链表的节点；一个指针在链表的前端进行移动。最后返回头结点的下一个节点即可。

1. 代码实现
2. C++

class Solution {

public:

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

//哨兵节点，更加方便最后返回合并后的链表

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;

}

//一个链表没有全部合并完

prev->next = l1 == nullptr ? l2 : l1;

return preHead->next;

}

};

1. Java

public class List21 {  
 public ListNode mergeTwoLists(ListNode l1, ListNode l2){  
 ListNode prehead = new ListNode(-1);  
 ListNode prev = prehead;  
 while(l1 != null && l2 != null){  
 if(l1.val <= l2.val){  
 prev.next = l1;  
 l1 = l1.next;  
 }  
 else{  
 prev.next = l2;  
 l2 = l2.next;  
 }  
 prev = prev.next;  
 }  
  
 prev.next = l1==null ? l2 : l1;  
 return prehead.next;  
 }  
}