# 题目

在 O(n log n) 时间复杂度和常数级空间复杂度下，对链表进行排序。

**示例 1:**

输入: 4->2->1->3

输出: 1->2->3->4

**示例 2:**

输入: -1->5->3->4->0

输出: -1->0->3->4->5

# 分析

## 方法一：归并排序

class Solution {

public:

ListNode \* sortList(ListNode \* head)

{

return (head == NULL)? NULL: mergeSort(head);

}

private:

ListNode \* findMid(ListNode \* head)

{

ListNode \* slow = head;

ListNode \* fast = head;

ListNode \* previous = NULL;

while (fast != NULL && fast->next != NULL)

{

previous = slow;

slow = slow->next;

fast = fast->next->next;

}

// split the list into two parts

previous->next = NULL;

return slow;

}

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;

}

}

ListNode \* mergeSort(ListNode \* head)

{

if (head->next == NULL) return head;

ListNode \* mid = findMid(head);

ListNode \* l1 = mergeSort(head);

ListNode \* l2 = mergeSort(mid);

return mergeTwoLists(l1, l2);

}

};