# 题目

给定链表的头结点 head ，请将其按 升序 排列并返回 排序后的链表 。

示例 1：



输入：head = [4,2,1,3]

输出：[1,2,3,4]

示例 2：



输入：head = [-1,5,3,4,0]

输出：[-1,0,3,4,5]

示例 3：

输入：head = []

输出：[]

提示：

链表中节点的数目在范围 [0, 5 \* 104] 内

-105 <= Node.val <= 105

进阶：你可以在 O(nlogn) 时间复杂度和常数级空间复杂度下，对链表进行排序吗？

注意：本题与主站 148 题相同：https://leetcode-cn.com/problems/sort-list/

# 分析

/\*\*

 \* 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\* sortList(ListNode\* head) {

        return sort(head,nullptr);

    }

    ListNode \*sort(ListNode \*head,ListNode \*tail)

    {

        if(nullptr==head)

            return head;

        if(head->next==tail)

        {

            head->next = nullptr;

            return head;

        }

        ListNode \*slow=head,\*fast=head;

        while(fast!=tail)

        {

            slow = slow->next;

            fast = fast->next;

            if(fast!=tail)

                fast = fast->next;

        }

        ListNode \*mid = slow;

        return merge(sort(head,mid),sort(mid,tail));

    }

    ListNode \*merge(ListNode \*head1,ListNode \*head2)

    {

        ListNode \*dummyNode = new ListNode(0);

        ListNode \*tmpNode = dummyNode;

        ListNode \*p=head1,\*q=head2;

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

        {

            if(p->val <= q->val)

            {

                tmpNode->next = p;

                p = p->next;

            }

            else

            {

                tmpNode->next = q;

                q = q->next;

            }

            tmpNode = tmpNode->next;

        }

        if(nullptr!=p)

        {

            tmpNode->next = p;

        }

        else if(nullptr!=q)

        {

            tmpNode->next = q;

        }

        return dummyNode->next;

    }

};