```
연결 리스트를 삽입 정렬로 정렬하라.
Input: head = [1,2,3,4]
Output: [1,4,2,3]
# Definition for singly-linked list.
# class ListNode:
   def __init__(self, val=0, next=None):
#
      self.val = val
#
      self.next = next
1.
class Solution:
  def insertionSortList(self, head: ListNode) -> ListNode:
    def delNode(node):
       temp = head
       while temp.next != node:
         temp = temp.next
       temp.next = temp.next.next
    def insNode(node_ori, node_ins):
       node_ori.next, node_ins.next = node_ins, node_ori.next
    head = ListNode(next = head)
    temp = head.next
    while temp != None:
       next = temp.next
       left = head
       while left.next != temp and left.next.val < temp.val:
         left = left.next
       delNode(temp)
       insNode(left, temp)
       temp = next
    return head.next
class Solution:
  def insertionSortList(self, head: ListNode) -> ListNode:
    cur = parent = ListNode(None)
    while head:
       while cur.next and cur.next.val < head.val:
         cur = cur.next
       cur.next, head.next, head = head, cur.next, head.next
       cur = parent
     return cur.next
```

class Solution:

```
def insertionSortList(self, head: ListNode) -> ListNode:
    cur = parent = ListNode(0)
    while head:
        while cur.next and cur.next.val < head.val:
            cur = cur.next
            cur.next, head.next, head = head, cur.next, head.next
        if head and cur.val > head.val:
            cur = parent
    return parent.next
```