# Definition for singly-linked list.

# class ListNode:

# def \_\_init\_\_(self, val=0, next=None):

# self.val = val

# self.next = next

class Solution:

def mergeTwoLists(self, list1: Optional[ListNode], list2: Optional[ListNode]) -> Optional[ListNode]:

# Create a dummy node to simplify handling the head of the merged list.

dummy = ListNode()

# 'tail' will always point to the last node of the merged list.

tail = dummy

# Iterate while both lists have nodes.

while list1 and list2:

if list1.val < list2.val:

tail.next = list1

list1 = list1.next

else:

tail.next = list2

list2 = list2.next

# Move the 'tail' pointer to the newly added node.

tail = tail.next

# If one list is exhausted, append the remaining nodes of the other list.

if list1:

tail.next = list1

elif list2:

tail.next = list2

# Return the head of the merged list, which is the node after the dummy node.

return dummy.next