41. You are given the heads of two sorted linked lists list1 and list2.

Merge the two lists in a one sorted list. The list should be made by splicing together the

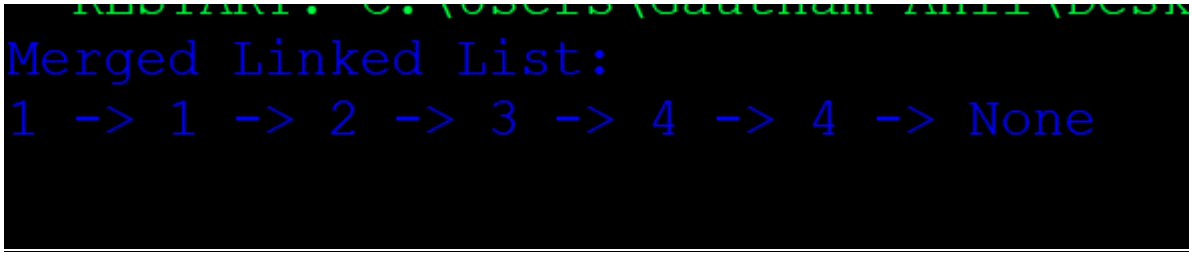
nodes of the first two lists.

Return the head of the merged linked list.

Code:

```
class ListNode:
    def __init__(self, val=0, next=None):
        self.val = val
        self.next = next
def mergeTwoLists(list1, list2):
    dummy = ListNode()
    current = dummy

    while list1 and list2:
        if list1.val < list2.val:
            current.next = list1
            list1 = list1.next
        else:
            current.next = list2
            list2 = list2.next
        current = current.next

    if list1:
        current.next = list1
    else:
        current.next = list2

    return dummy.next

def printList(node):
    while node:
        print(node.val, end=" -> ")
        node = node.next
    print("None")
def createList(arr):
    if not arr:
        return None
    head = ListNode(arr[0])
    current = head
    for value in arr[1:]:
        current.next = ListNode(value)
        current = current.next
    return head
list1 = createList([1, 2, 4])
list2 = createList([1, 3, 4])
mergedList = mergeTwoLists(list1, list2)
print("Merged Linked List:")
printList(mergedList)
```

Output:



Time Complexity:

- T(n)= O(nlogk)