class Solution {

public:

ListNode\* mergeTwoLists(ListNode\* list1, ListNode\* list2) {

if (list1 == nullptr) {

return list2;

}

if (list2 == nullptr) {

return list1;

}

ListNode\* dummy = new ListNode(); // Dummy node to simplify the code

ListNode\* current = dummy;

while (list1 != nullptr && list2 != nullptr) {

if (list1->val <= list2->val) {

current->next = list1;

list1 = list1->next;

} else {

current->next = list2;

list2 = list2->next;

}

current = current->next;

}

// Add any remaining nodes from list1 or list2

if (list1 != nullptr) {

current->next = list1;

}

if (list2 != nullptr) {

current->next = list2;

}

ListNode\* mergedHead = dummy->next;

delete dummy; // Clean up the dummy node

return mergedHead;

}

};