Description | Editorial | Solutions | Submissions

21. Merge Two Sorted Lists

Solved

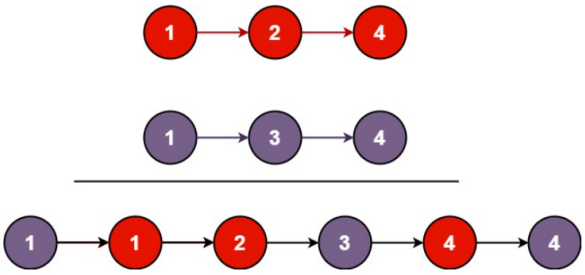
Easy | Topics | Companies

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

Merge the two lists into 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.*

Example 1:




```
Input: list1 = [1,2,4], list2 = [1,3,4]
Output: [1,1,2,3,4,4]
```

Example 2:

```
Input: list1 = [], list2 = []
Output: []
```

Example 3:

```
Input: list1 = [], list2 = [0]
Output: [0]
```

Constraints:

- The number of nodes in both lists is in the range `[0, 50]`.
- `-100 <= Node.val <= 100`
- Both `list1` and `list2` are sorted in **non-decreasing** order.

Seen this question in a real interview before? 1/5

Yes | No

Accepted 4.4M | Submissions 6.8M | Acceptance Rate 65.0%

Topics

Companies

Similar Questions

Discussion (302)

21.9K | 302

Code

C++ | Auto

```cpp
 * };
 */
class Solution {
public:
    ListNode* mergeTwoLists(ListNode* head1, ListNode* head2) {

        if (head1 == NULL)
            return head2;
        if (head2 == NULL)
            return head1;

        ListNode* ans = new ListNode(-1);
        ListNode* ptr = ans;

        while (head1 != NULL and head2 != NULL) {

            if (head1->val <= head2->val) {
                ptr->next = head1;
                ptr = head1;
                head1 = head1->next;
            } else {
                ptr->next = head2;
                ptr = head2;
                head2 = head2->next;
            }
        }

        if (head1 != NULL) {
            ptr->next = head1;
        }

        if (head2 != NULL) {
            ptr->next = head2;
        }
        return ans->next;
    }
};
```

Saved | Ln 1, Col 1

Testcase | Test Result