21. Merge Two Sorted Lists

Solved

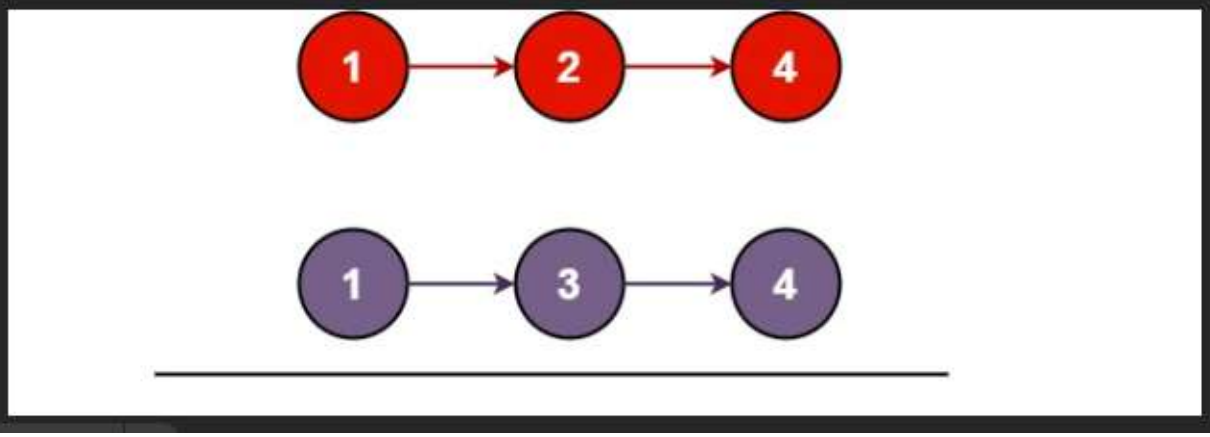
Easy | Topics | Companies

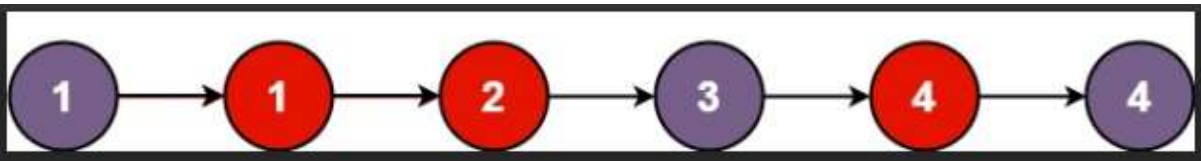
You are given the heads of two sorted linked lists `list1` and `list2`.

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:** []

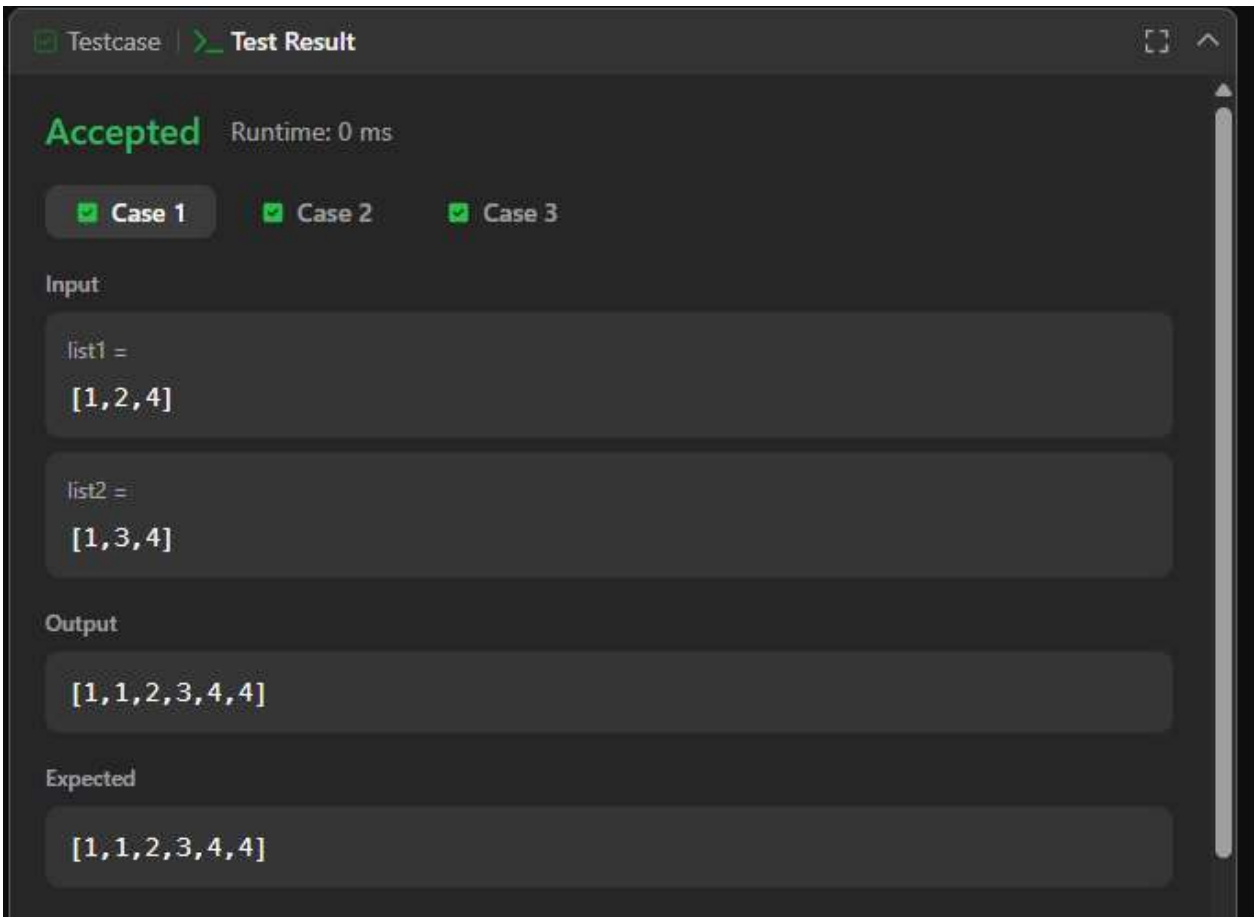
Example 3:

> **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.

```c
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
struct ListNode* mergeTwoLists(struct ListNode* list1, struct ListNode* list2) {
    if (list1 == NULL)
        return list2;
    if (list2 == NULL)
        return list1;
    struct ListNode *temp=list1;
    while(temp->next!=NULL){
        temp=temp->next;
    }
    temp->next=list2;
    int swap;
    struct ListNode *ptr1;
    struct ListNode *lptr=NULL;
    do{
        swap=0;
        ptr1=list1;
        while(ptr1->next!=lptr){
            if(ptr1->val > ptr1->next->val){
                int c=ptr1->val;
                ptr1->val=ptr1->next->val;
                ptr1->next->val=c;
                swap=1;
            }
            ptr1=ptr1->next;
        }
        lptr=ptr1;
    }while(swap);
    return list1;
}
```


Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
list1 =
[1,2,4]
list2 =
[1,3,4]
Output
[1,1,2,3,4,4]
Expected
[1,1,2,3,4,4]