

# LEETCODE SOLUTIONS - YOGESH MUNEEES

## 21. Merge Two Sorted Lists

Easy



19.6K

1.8K



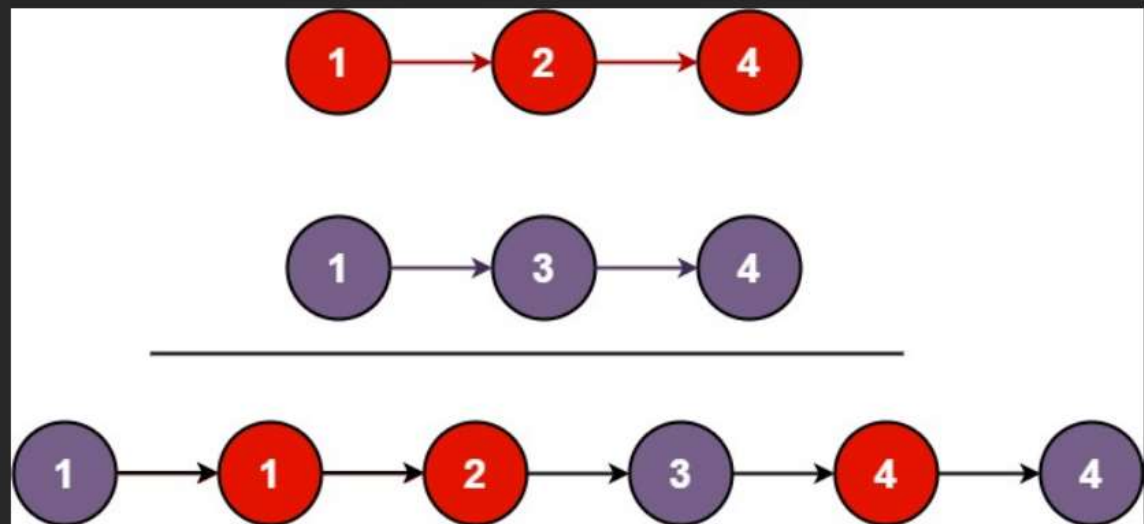
Companies

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:**



**Solution:**

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
struct ListNode* mergeTwoLists(struct ListNode* list1, struct ListNode* list2){

    struct ListNode * temp1 = list1;
    struct ListNode * temp2 = list2;

    if(temp2==NULL) return list1;
    if(temp1==NULL) return list2;
```

```

struct ListNode * add = (struct ListNode *)malloc(sizeof(struct ListNode));
struct ListNode * track = add;

if(temp1->val <= temp2->val){
    add->val = temp1->val;
    add->next = NULL;
    temp1 = temp1->next;
} else {
    add->val = temp2->val;
    add->next = NULL;
    temp2 = temp2->next;
}

while(temp1!=NULL && temp2!=NULL){
    struct ListNode *new = (struct ListNode *)malloc(sizeof(struct ListNode));
    printf("%d %d\n",temp1,temp2->val);
    if(temp1->val <= temp2->val){
        new->val = temp1->val;
        new->next = add->next;
        add->next = new;
        add = add->next;
        temp1 = temp1->next;
    }
    else{
        new->val = temp2->val;
        new->next = add->next;
        add->next = new;
        add = add->next;
        temp2 = temp2->next;
    }
}

if(temp1==NULL){
    add->next = temp2;
    return track;
}

if(temp2==NULL){
    add->next = temp1;
    return track;
}

return track;
}

```