143. Reorder List

DescriptionHintsSubmissionsSolutions

- Total Accepted: **91343**
- Total Submissions: **361499**
- Difficulty: **Medium**
- Contributor: **LeetCode**

Given a singly linked list *L*: *L*0→*L*1→…→*L*n-1→*L*n,

reorder it to: *L*0→*L*n→*L*1→*L*n-1→*L*2→*L*n-2→…

You must do this in-place without altering the nodes' values.

For example,

Given {1,2,3,4}, reorder it to {1,4,2,3}.

Subscribe to see which companies asked this question.

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    ListNode *reverse(ListNode* head)
    {
        ListNode *prev = head;
        ListNode *cur = head->next;
        ListNode *post;
        prev->next = NULL;
        while(cur)
        {
            post = cur->next;
            cur->next = prev;
            prev = cur;
            cur = post;
        }
        return prev;
    }

    void merge(ListNode* l1, ListNode* l2)
    {
        ListNode *cur1 = l1;
        ListNode *cur2 = l2;
        ListNode *post1,*post2;
        while(cur2)
        {
            post1 = cur1->next;
            post2 = cur2->next;
            cur1->next = cur2;
            cur2->next = post1;
            cur1 = post1;
            cur2 = post2;
        }
    }

    void reorderList(ListNode* head) {
        if(head==NULL || head->next ==NULL || head->next->next == NULL) return;
        ListNode *fast = head;
        ListNode *slow = head;
        while(fast->next!=NULL && fast->next->next!=NULL)
        {
            fast = fast->next->next;
            slow = slow->next;
        }
        ListNode *head2 = slow->next;
        slow->next = NULL;
        merge(head,reverse(head2));

    }
};
```