

328. Odd Even Linked List

Medium

Topics

Companies

Given the `head` of a singly linked list, group all the nodes with odd indices together followed by the nodes with even indices, and return *the reordered list*.

The **first** node is considered **odd**, and the **second** node is **even**, and so on.

Note that the relative order inside both the even and odd groups should remain as it was in the input.

You must solve the problem in $O(1)$ extra space complexity and $O(n)$ time complexity.

```
struct ListNode* oddEvenList(struct ListNode* head)
{
    if (head == NULL || head->next == NULL)
    {
        return head;
    }
    struct ListNode* oddTemp = head;
    struct ListNode* evenTemp = head->next;
    struct ListNode* evenHead = evenTemp;
    while (evenTemp != NULL && evenTemp->next != NULL)
    {
        oddTemp->next = evenTemp->next;
        oddTemp = oddTemp->next;
        evenTemp->next = oddTemp->next;
        evenTemp = evenTemp->next;
    }
    oddTemp->next = evenHead;
    return head;
}
```

Accepted Runtime: 2 ms

- Case 1
- Case 2

Input

head =
[1,2,3,4,5]

Output

[1,3,5,2,4]

Expected

[1,3,5,2,4]

Accepted Runtime: 2 ms

- Case 1
- Case 2

Input

head =
[2,1,3,5,6,4,7]

Output

[2,3,6,7,1,5,4]

Expected

[2,3,6,7,1,5,4]

NA
26/2/24