Problem 328: Odd Even Linked List

Problem Information

Difficulty: Medium
Acceptance Rate: 62.16%
Paid Only: No
Tags: Linked List

Problem Description

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.

Example 1:

![](https://assets.leetcode.com/uploads/2021/03/10/oddeven-linked-list.jpg)

Input: head = [1,2,3,4,5] **Output:** [1,3,5,2,4]

Example 2:

![](https://assets.leetcode.com/uploads/2021/03/10/oddeven2-linked-list.jpg)

Input: head = [2,1,3,5,6,4,7] **Output:** [2,3,6,7,1,5,4]

Constraints:

* The number of nodes in the linked list is in the range `[0, 104]`. * `-106 <= Node.val <= 106`

Code Snippets

C++:

```
/**
* Definition for singly-linked list.
* struct ListNode {
* int val;
* ListNode *next;
* ListNode() : val(0), next(nullptr) {}
* ListNode(int x) : val(x), next(nullptr) {}
* ListNode(int x, ListNode *next) : val(x), next(next) {}
* };
*/
class Solution {
public:
ListNode* oddEvenList(ListNode* head) {

}
};
```

Java:

```
/**
* Definition for singly-linked list.
* public class ListNode {
* int val;
* ListNode next;
* ListNode() {}
* ListNode(int val) { this.val = val; }
* ListNode(int val, ListNode next) { this.val = val; this.next = next; }
* }
*/
class Solution {
public ListNode oddEvenList(ListNode head) {

}
}
```

Python3:

```
# Definition for singly-linked list.
# class ListNode:
# def __init__(self, val=0, next=None):
# self.val = val
# self.next = next
class Solution:
def oddEvenList(self, head: Optional[ListNode]) -> Optional[ListNode]:
```