0143. 重排链表

ITCharge 大约 1 分钟

- 标签：栈、递归、链表、双指针
- 难度：中等

题目链接

- 0143. 重排链表 - 力扣

题目大意

描述：给定一个单链表 L 的头节点 $head$，单链表 L 表示为：$L_0 \rightarrow L_1 \rightarrow L_2 \rightarrow ... \rightarrow L_{n-1} \rightarrow L_n$。

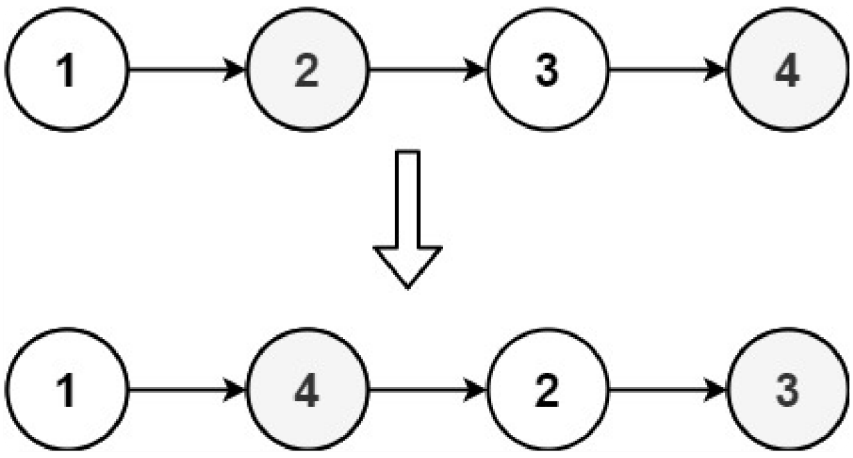
要求：将单链表 L 重新排列为：$L_0 \rightarrow L_n \rightarrow L_1 \rightarrow L_{n-1} \rightarrow L_2 \rightarrow L_{n-2} \rightarrow L_3 \rightarrow L_{n-3} \rightarrow \cdots$。

说明：

- 需要将实际节点进行交换。

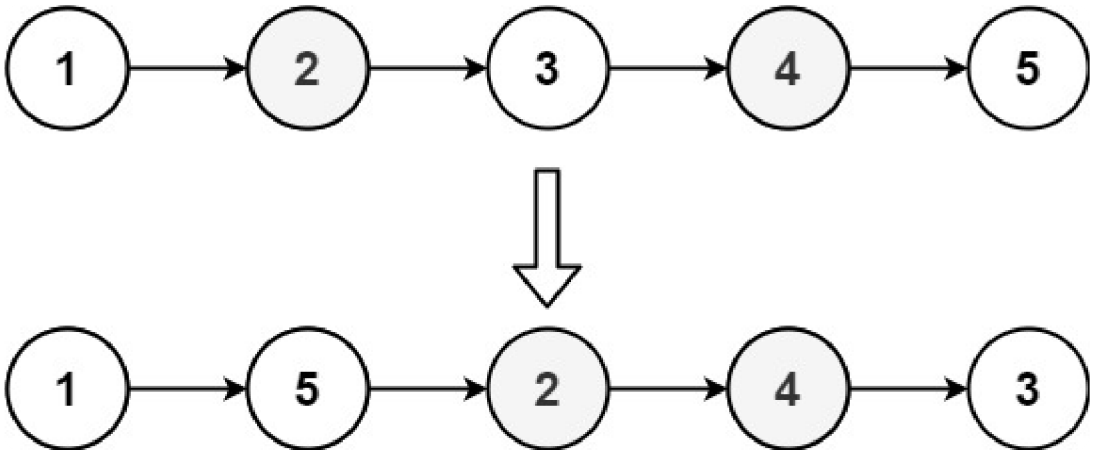
示例：

- 示例 1：



```py
输入：head = [1,2,3,4]
输出：[1,4,2,3]
```

- 示例 2：




```py
输入：head = [1,2,3,4,5]
输出：[1,5,2,4,3]
```

解题思路

思路 1：线性表

因为链表无法像数组那样直接进行随机访问。所以我们可以先将链表转为线性表，然后直接按照提要要求的排列顺序访问对应数据元素，重新建立链表。

思路 1：代码

```py
class Solution:
    def reorderList(self, head: ListNode) -> None:
        """
        Do not return anything, modify head in-place instead.
        """
        if not head:
            return

        vec = []
```

```
        node = head
        while node:
            vec.append(node)
            node = node.next

        left, right = 0, len(vec) - 1
        while left < right:
            vec[left].next = vec[right]
            left += 1
            if left == right:
                break
            vec[right].next = vec[left]
            right -= 1
        vec[left].next = None
```

思路 1：复杂度分析

- **时间复杂度**：$O(n)$。
- **空间复杂度**：$O(n)$。