

24. 两两交换链表中的节点**

地址：[🔗 两两交换链表中的节点](#)

题目：

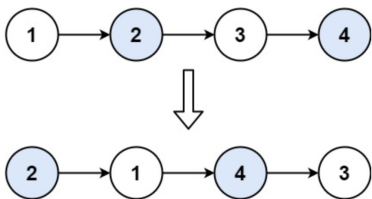
- English：

24. Swap Nodes in Pairs

难度 中等 966 收藏 分享 讨论 标记

Given a linked list, swap every two adjacent nodes and return its head. You must solve the problem without modifying the values in the list's nodes (i.e., only nodes themselves may be changed.)

Example 1:



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

Example 2:

Input: head = []
Output: []

Example 3:

Input: head = [1]
Output: [1]

Constraints:

- The number of nodes in the list is in the range `[0, 100]`.
- `0 <= Node.val <= 100`

通过次数 278.793 提交次数 398.587

- 中文：

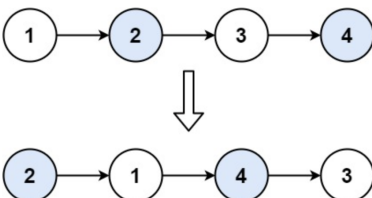
24. 两两交换链表中的节点

难度 中等 966 收藏 分享 讨论 标记

给定一个链表，两两交换其中相邻的节点，并返回交换后的链表。

你不能只是单纯的改变节点内部的值，而是需要实际的进行节点交换。

示例 1：



输入: head = [1,2,3,4]
输出: [2,1,4,3]

示例 2：

输入: head = []
输出: []

示例 3：

输入: head = [1]
输出: [1]

提示：

- 链表中节点的数目在范围 `[0, 100]` 内
- `0 <= Node.val <= 100`

进阶：你能在不修改链表节点值的情况下解决这个问题吗？（也就是说，仅修改节点本身。）

思路 1: 双指针之快慢指针

分析

★ 双指针之快慢指针：

★ slow pointer：从列表头开始；

★ fast pointer：从列表头后一个节点开始。

★ 遍历过程：

★ 每次遍历将快慢指针的值互换；

★ 快慢指针每次都过两个节点。

★ 遍历结束条件：

★ fast.next == null：节点数为双数时，以防越界；

★ fast.next.next == null：节点数为单数时，以防越界。

★ 返回 head list.

代码：

```
1 // Java
2 // Time : 2021 - 07 - 12
3
4
5 public ListNode swapPairs(ListNode head) {
6
7
8     // list empty or single node
9     if (head == null || head.next == null) return head;
10
11
12     ListNode slow = head;           // slow pointer
13     ListNode fast = head.next;      // fast pointer
14
15     // loop through the list
16     while (true) {
17
18         // swap the value between slow and fast node
19         int temp = fast.val;
20         fast.val = slow.val;
21         slow.val = temp;
22
23         // loop through end
24         if (fast.next == null || fast.next.next == null) break;
25
26         // move slow and fast pointer to next next node
27         else {
28             slow = slow.next.next;
29             fast = fast.next.next;
```

```
30     }
31 }
32
33 // return result list
34 return head;
35 }
```

复杂度分析：

- 时间复杂度： $O(n)$
- 空间复杂度： $O(1)$