极客学院
jikexueyuan.com

名企数据结构面试题之链表（下）

名企数据结构面试题之链表（下） — 课程概要

- 旋转链表
- 回文链表
- 交换链表的相邻节点
- 链表划分
- 链表洗牌

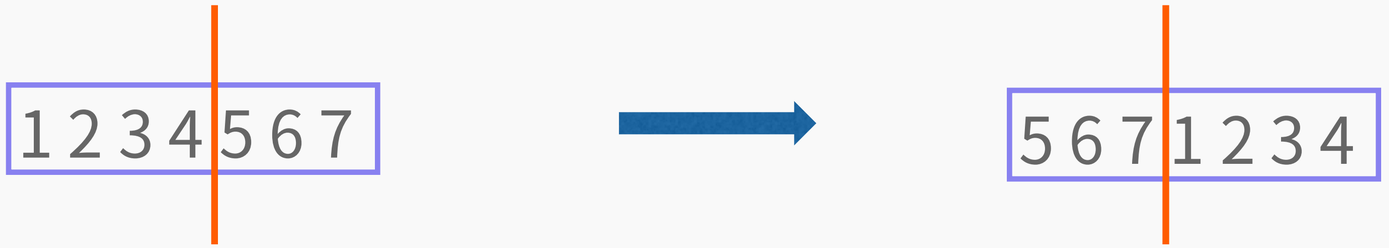
旋转链表

旋转链表

- 问题描述
- 思路分析
- 代码实现
- 测试与提交
- 课后练习

旋转链表 — 问题描述

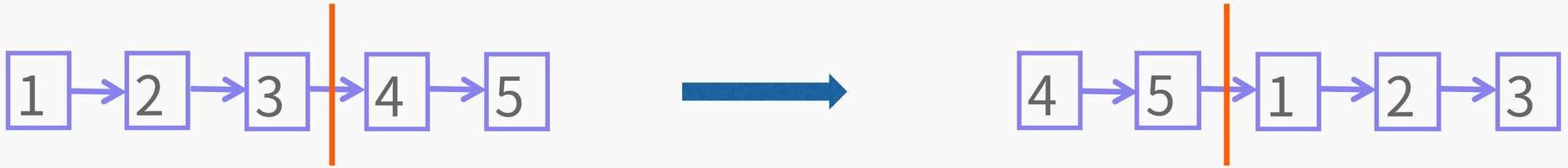
旋转数组：



旋转字符串：

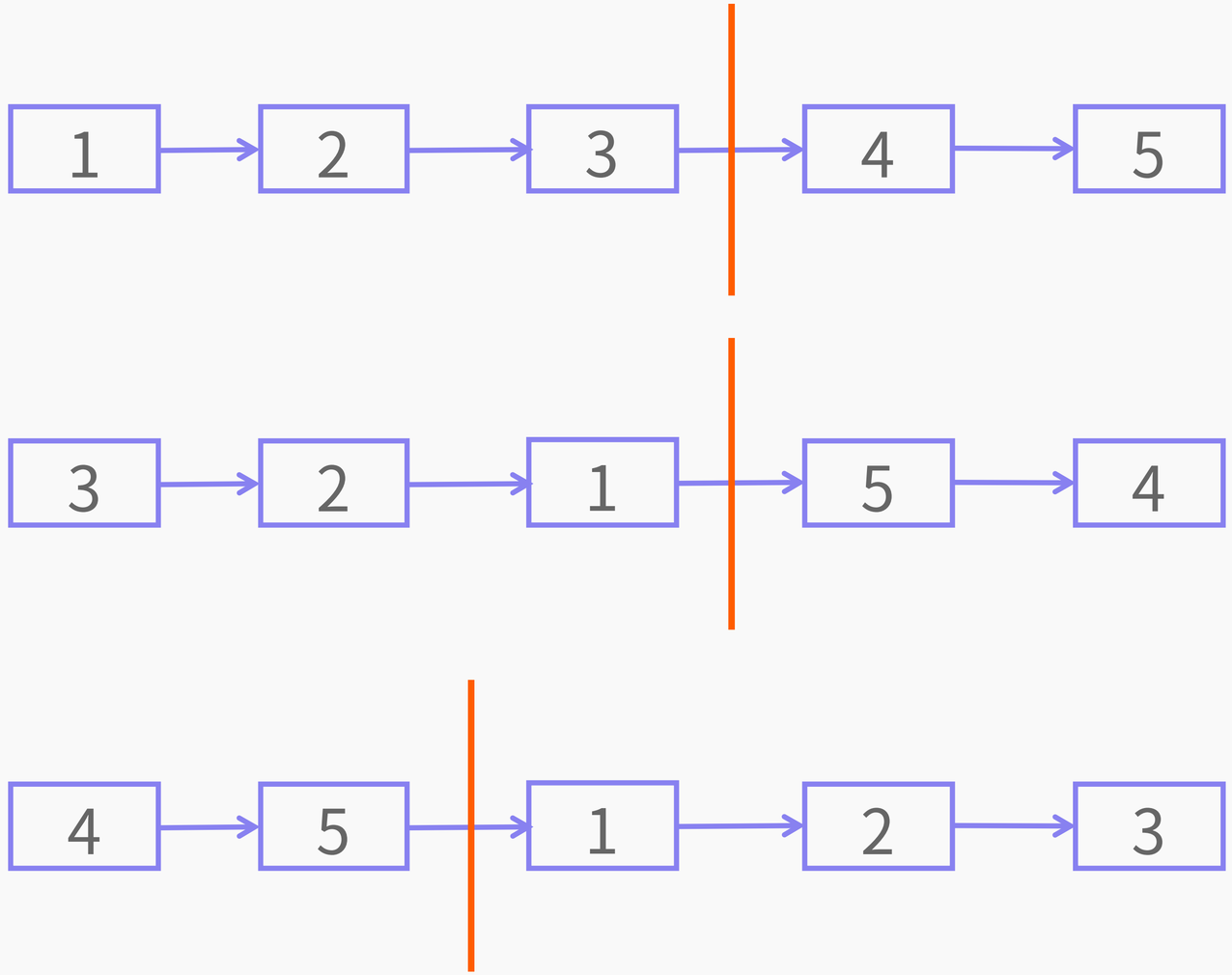
反转单词顺序：

leetCode 61：Rotate List（旋转链表）



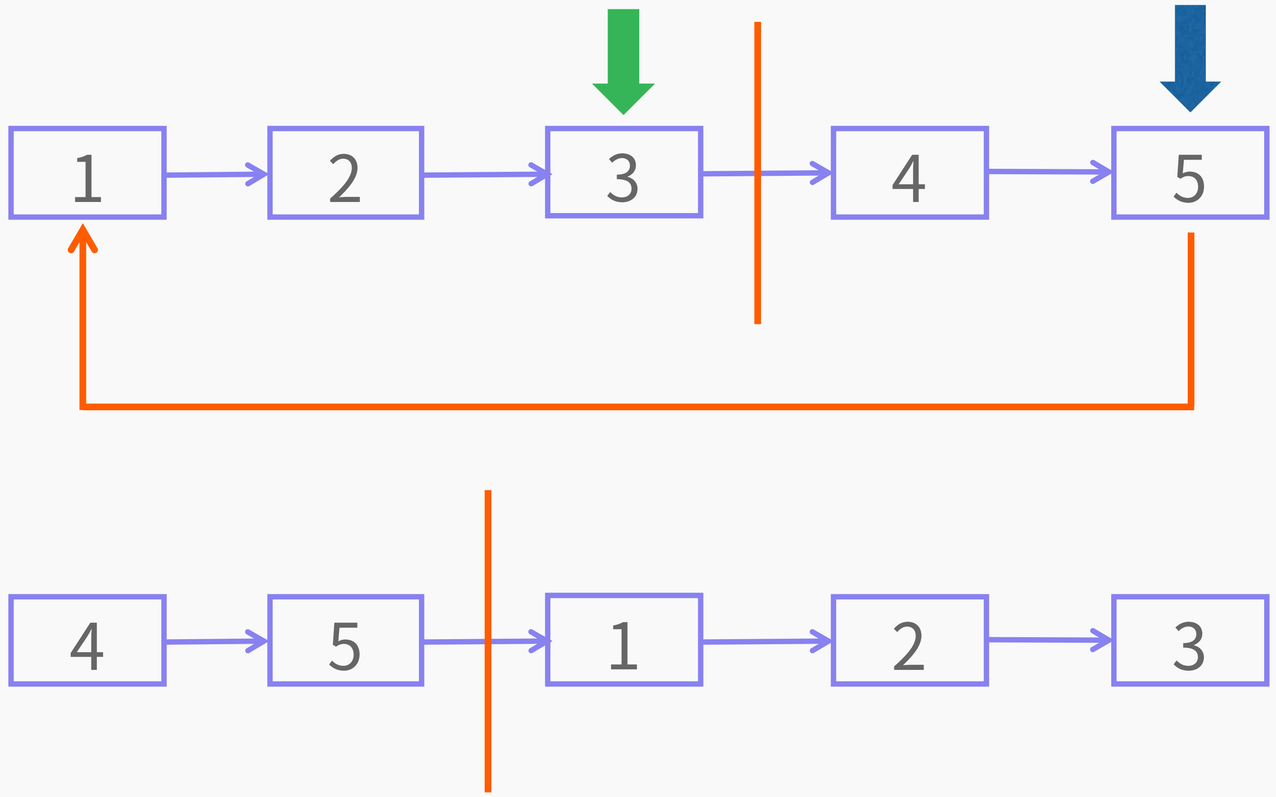
旋转链表 — 思路分析

思路1：

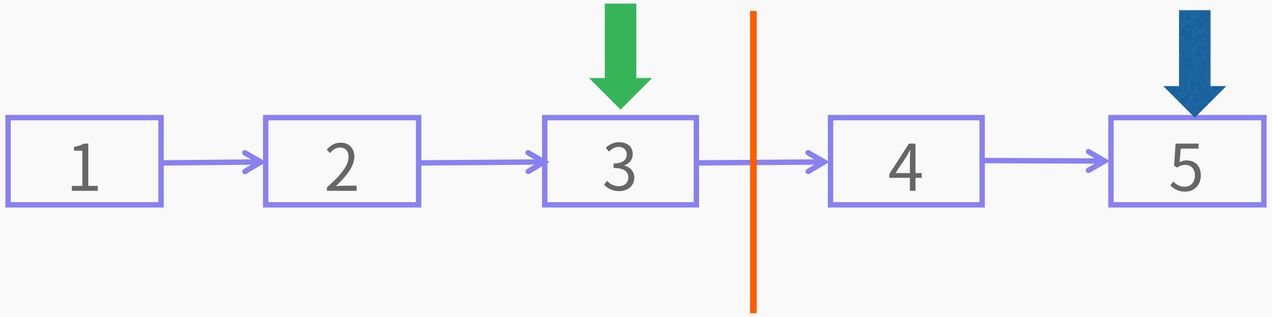


旋转链表 — 思路分析

思路2：



旋转链表 — 思路分析



寻找链表的倒数第k+1个节点

- 普通算法
- OnePass算法

k有可能大于链表长度n，k=k%n，比如：2=7%5

旋转链表 — 代码实现

Key	Value
类名	_061RotateList
方法名	rotateRight
时间复杂度	O(N)，N表示链表长度
空间复杂度	O(1)

旋转链表 — 测试与提交

Key	Value
类名	_061RotateList
方法名	test
测试输入1	1→2→3→4→5, k = 2
测试输出1	4→5→1→2→3
测试输入	1→2→3→4→5→6, k = 10
测试输出	3→4→5→6→1→2→3

旋转链表 — 课后练习

试一试：

- 分别反转、全体反转
- OnePass算法，有可能存在Bug

回文链表

回文链表

- 问题描述
- 思路分析
- 代码实现
- 测试与提交
- 其它注意点

回文链表 — 问题描述

回文字符串：

abcdcba

回文数字：

12344321

leetCode 234：Palindrome Linked List（回文链表）

回文链表 — **问题描述**

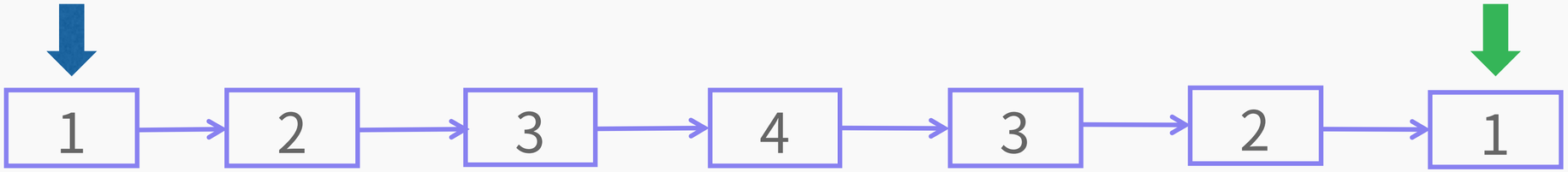
额外要求：

- 时间复杂度为O(N)，但不一定OnePass
- 空间复杂度为O(1)

回文链表 — 思路分析

回文判断：

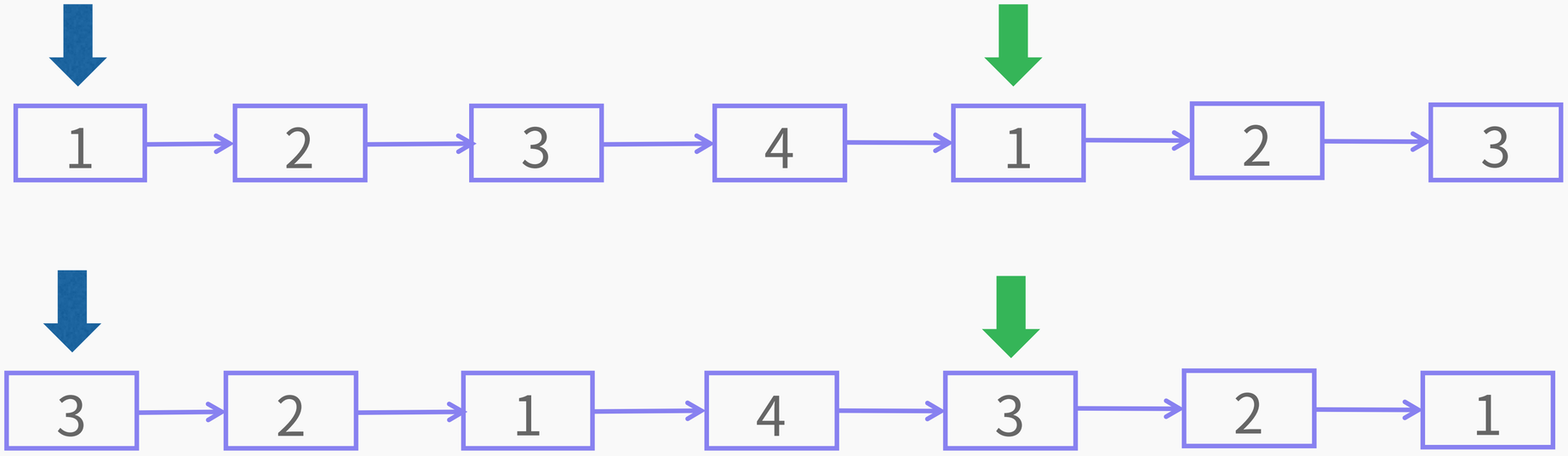
- 从两头往中间扫描
- 从中间往两头扫描



没有pre指针！

回文链表 — 思路分析

- 反转任意一半
- 双指针，从前往后扫描



注意点：链表长度的奇偶判断

回文链表 — 代码实现

Key	Value
类名	_234PalindromeLinkedList
方法名	isPalindrome
时间复杂度	O(N)，N表示链表长度
空间复杂度	O(1)

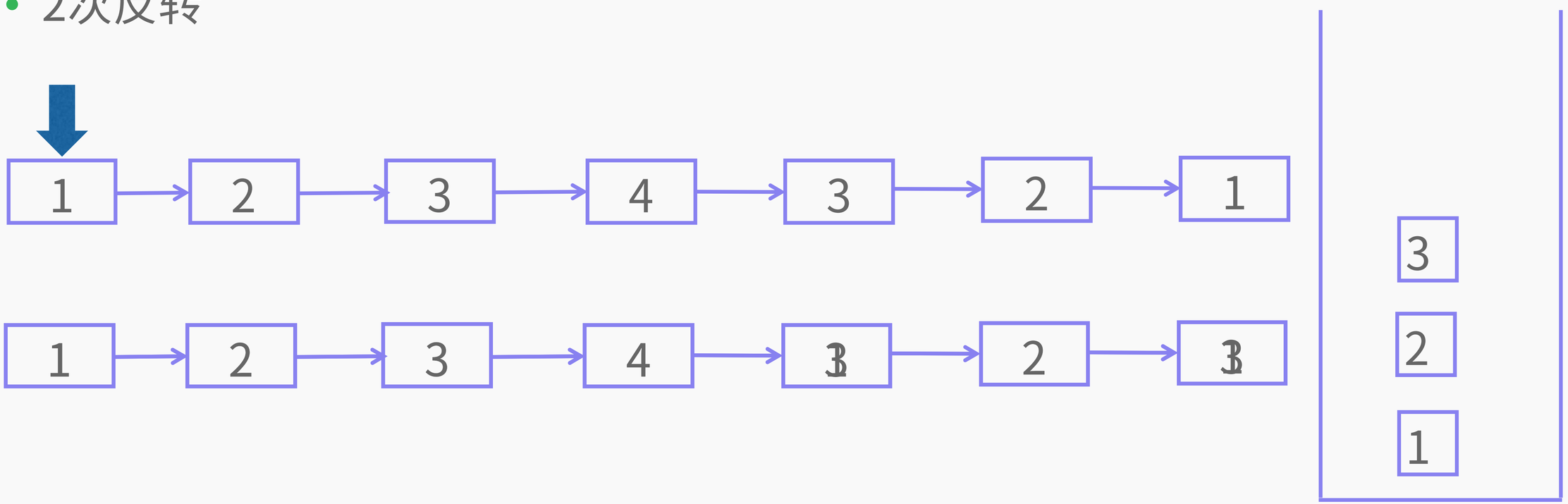
回文链表 — 测试与提交

Key	Value
类名	_234PalindromeLinkedList
方法名	test
测试输入1	1→2→3→4→3→2→1
测试输出1	true
测试输入2	1→2→3→4→1→2→3
测试输出2	false

回文链表 — **其它注意点**

不允许改变链表结构

- 栈
- 2次反转



交换链表的相邻节点

交换链表的相邻节点

- 问题描述
- 思路分析
- 代码实现
- 测试与提交

交换链表的相邻节点 — **问题描述**

leetCode 24：Swap Nodes in Pairs

样例输入：1→2→3→4→5→6

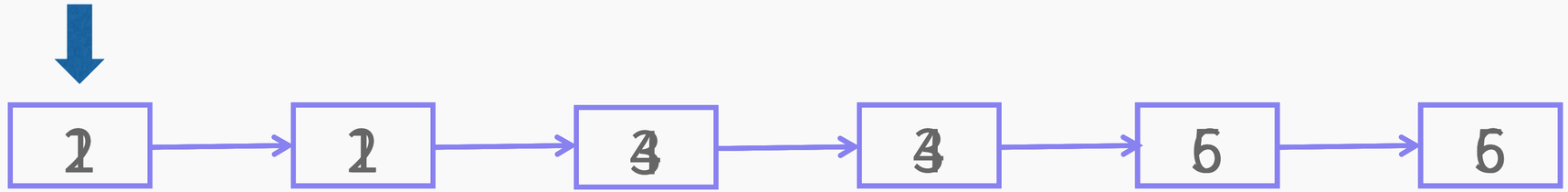
样例输出：2→1→4→3→6→5

额外要求：

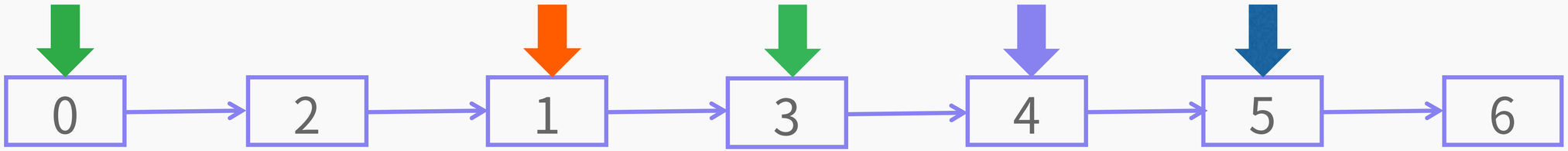
- 空间复杂度为O(1)
- 时间复杂度为O(N)
- 不允许改变节点的值

交换链表的相邻节点 — **思路分析**

遍历链表，每次反转两个节点



需要多少个指针?



交换链表的相邻节点 — **代码实现**

Key	Value
类名	_024SwapNodesInPairs
方法名	swapPairs
时间复杂度	O(N)，N表示链表长度
空间复杂度	O(1)

交换链表的相邻节点 — 测试与提交

Key	Value
类名	_024SwapNodesInPairs
方法名	test
测试输入1	1→2→3→4→5→6
测试输出1	2→1→4→3→6→5
测试输入2	1→2→3→4→5
测试输出2	2→1→4→3→5

链表划分

链表划分

- 初识Partition
- 问题描述
- 思路分析
- 代码实现
- 测试与提交
- 其它知识点

链表划分 — 初识Partition

Partition（划分）的概念：

对于某个线性结构，通过某种操作（算法），使得它的左半部分具有某种属性，右半部分具有另外的属性，这种操作（算法）叫做**Partition**。

例如**快速排序**的一次划分：

选取随机元素为枢纽元**pivot**，使得数组的左边的元素都小于pivot，右边的元素大于或等于pivot。

链表划分 — **问题描述**

leetCode 86：Partition List

给定链表和值x，划分链表，将小于x的节点放到左边，大于等于x的节点放到右边。

样例输入：1→4→3→2→5→2，x=3

样例输出：1→2→2→4→3→5

额外条件：

- 时间复杂度为O(N)
- 空间复杂度为O(1)
- 保持节点的**自然顺序**不变

链表划分 — 问题描述

自然顺序不变

样例输入：1→4→3→2→5→2，x=3

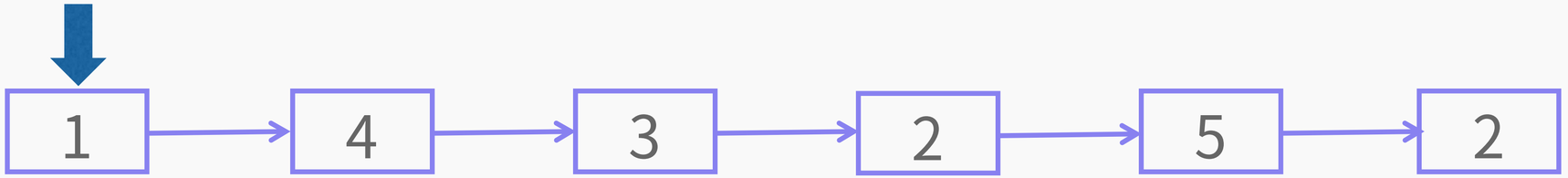
正确样例输出：1→2→2→4→3→5

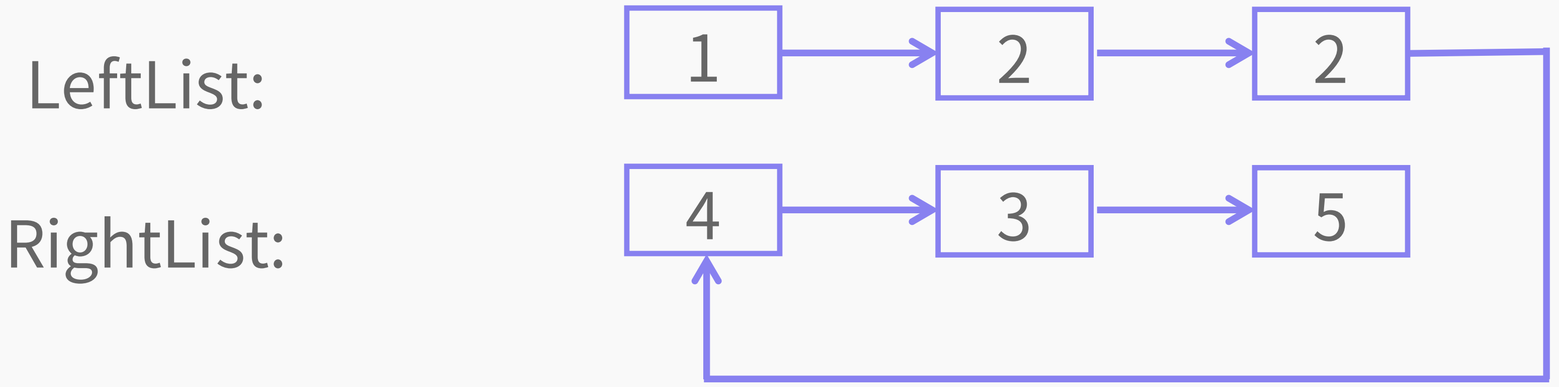
错误样例输出1：2→1→2→5→3→4

错误样例输出2： 1→2→2→4→3→5

链表划分 — 思路分析

- 新建左链表LeftList、右链表RightList
- 遍历链表，小于x的节点插入LeftList的尾部，大于等于x的节点同理
- 合并LeftList和RightList，将LeftList的尾节点与RightList的头节点相连
- 返回LeftList的头结点，作为结果





链表划分 — **思路分析**

注意点：

- 如何实现**尾部插入**
- 是否需要新建默认节点

链表划分 — 代码实现

Key	Value
类名	_086PartitionList
方法名	partition
时间复杂度	O(N)，N表示链表长度
空间复杂度	O(1)

链表划分 — 测试与提交

Key	Value
类名	_086PartitionList
方法名	test
测试输入	1→4→3→2→5→2
测试输出	1→2→2→4→3→5

链表划分 — **其它知识点**

有关Partition的其它面试题：

- 快速排序
- 奇偶数分离
- 正负数分割
- 荷兰国旗问题
- 回文子串的划分

以后再探讨！

链表划分 — **其它知识点**

自然顺序与排序算法的**稳定性**

稳定排序	不稳定排序
冒泡排序	选择排序
插入排序	希尔排序
归并排序	快速排序
基数排序	堆排序

以后再探讨！

链表洗牌

链表洗牌

- 问题描述
- 思路分析
- 代码实现
- 测试与提交
- 综合题小结

链表洗牌 — 问题描述

A	2	3	4	5	6	7	8	9	10	J	Q	K

A	8	2	9	3	10	4	J	5	Q	6	K	7

链表洗牌 — 问题描述

leetCode 143：Reorder List

给定单链表，实现反转与洗牌两个操作。

样例输入：1→2→3→4→5→6

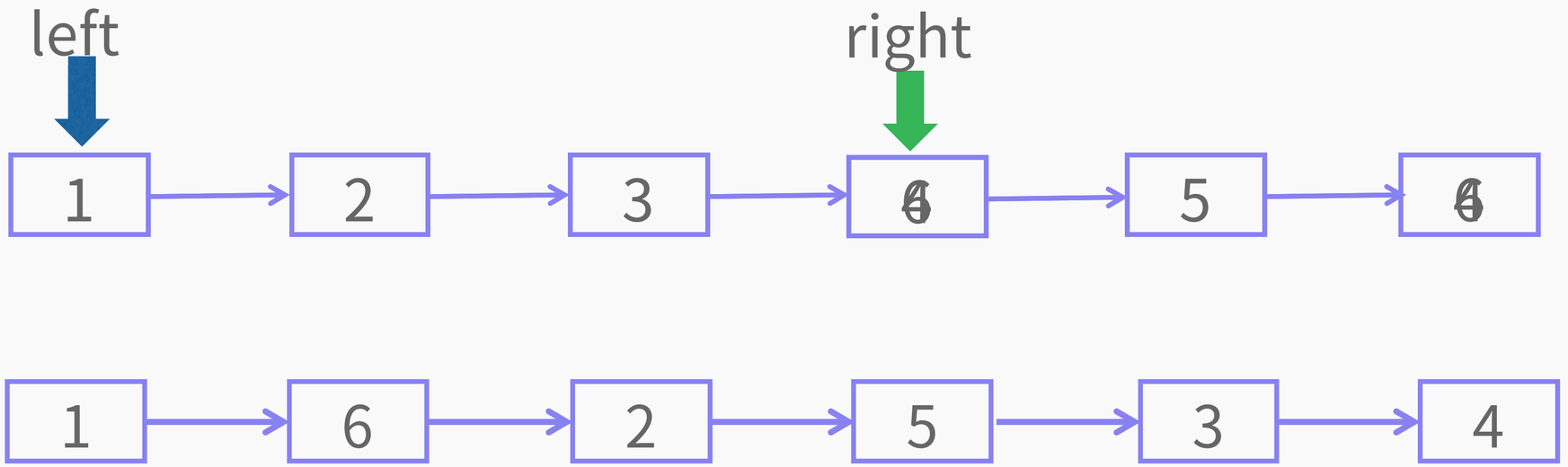
样例输出：1→6→2→5→3→4

额外要求：

- 空间复杂度为O(1)
- 不允许改变节点的值

链表洗牌 — 思路分析

- 取得链表长度
- 反转后一半
- 洗牌



链表洗牌 — **思路分析**

洗牌伪代码：

```
init(left, right);
flag=true;
next=null;
 while(right!=null){
      if(flag){
           next=left.next;
           left.next=right;
           left=next;
      }else    //相似操作
      flag=!flag;
}
```

链表洗牌 — 代码实现

Key	Value
类名	_143ReorderList
方法名	reorderList
时间复杂度	O(N)，N表示链表长度
空间复杂度	O(1)

链表洗牌 — 测试与提交

Key	Value
类名	_143ReorderList
方法名	test
测试输入1	1→2→3→4→5→6
测试输出1	1→6→2→5→3→4
测试输入2	1→2→3→4→5→6→7
测试输出2	1→7→2→6→3→5→4

链表洗牌 — **综合题小结**

- 空间复杂度为O(1)，不要轻易使用ArrayList、LinkedList、HashMap等等容器
- 时间复杂度为O(N)，LinearTime、OnePass
- 链表长度、链表反转、新建节点等等常用策略
- 旋转、回文、交换、划分、洗牌等等趣味操作

严谨认真、灵活机制、沟通交流

名企数据结构面试题之链表（下）

本套课程中我们学习了名企数据结构面试之链表（下）。你应当解决了以下面试题：

- 旋转链表
- 回文链表
- 交换链表的相邻节点
- 链表划分
- 链表洗牌

你可以使用leetCode验证程序是否正确，还可以在白纸上书写代码；如果想进一步提高，你可以继续在极客学院学习**链表环与链表交点**课程。

极客学院

jikexueyuan.com

中国最大的IT职业在线教育平台