

链表 (LinkedList)

@M了个J

<https://github.com/CoderMJLee>

<https://space.bilibili.com/325538782>



实力IT教育 www.520it.com





一杯茶，一包烟，一道链表做一天

👍 31 🗨 踩 💬 4 ↩ 回复

■ 解题建议

□ 一定要多画图

■ 解题技巧

□ 虚拟头结点

□ 快慢指针

□ 多指针

□

■ 常用代码要非常熟练

□ 链表节点的插入、删除

□ 反转（翻转）一个链表

□ 快慢指针求中心节点

□ 计算链表的长度

□

203. 移除链表元素

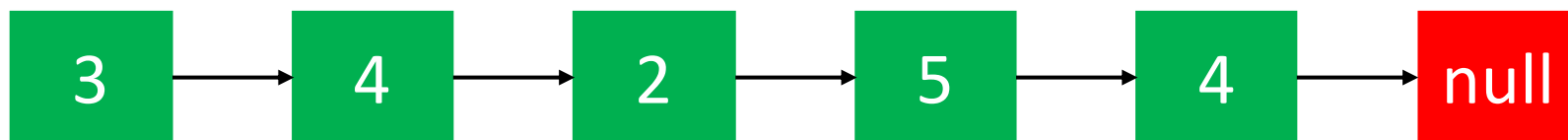
删除链表中等于给定值 *val* 的所有节点。

示例:

输入: 1->2->6->3->4->5->6, *val* = 6

输出: 1->2->3->4->5

■ 时间复杂度: $O(n)$ 、空间复杂度: $O(1)$



2. 两数相加

给出两个 **非空** 的链表用来表示两个非负的整数。其中，它们各自的位数是按照 **逆序** 的方式存储的，并且它们的每个节点只能存储 **一位** 数字。

如果，我们将这两个数相加起来，则会返回一个新的链表来表示它们的和。

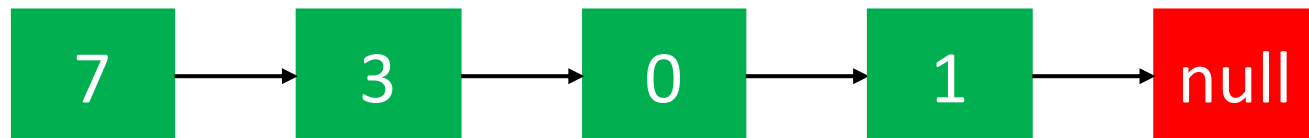
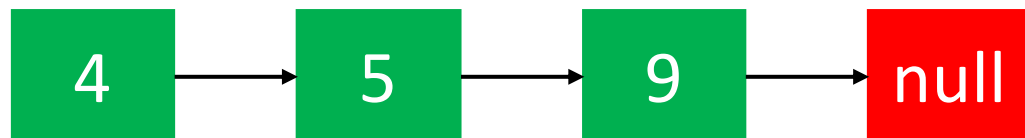
您可以假设除了数字 0 之外，这两个数都不会以 0 开头。

示例：

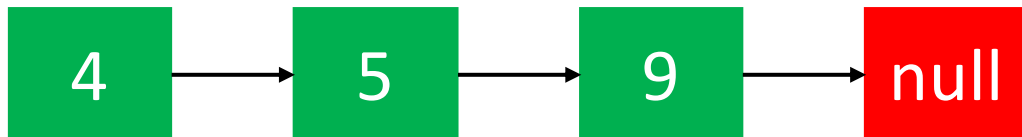
输入：(2 -> 4 -> 3) + (5 -> 6 -> 4)

输出：7 -> 0 -> 8

原因：342 + 465 = 807



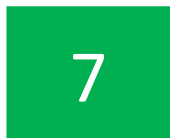
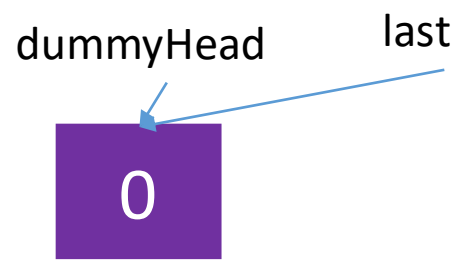
l1



进位: 0



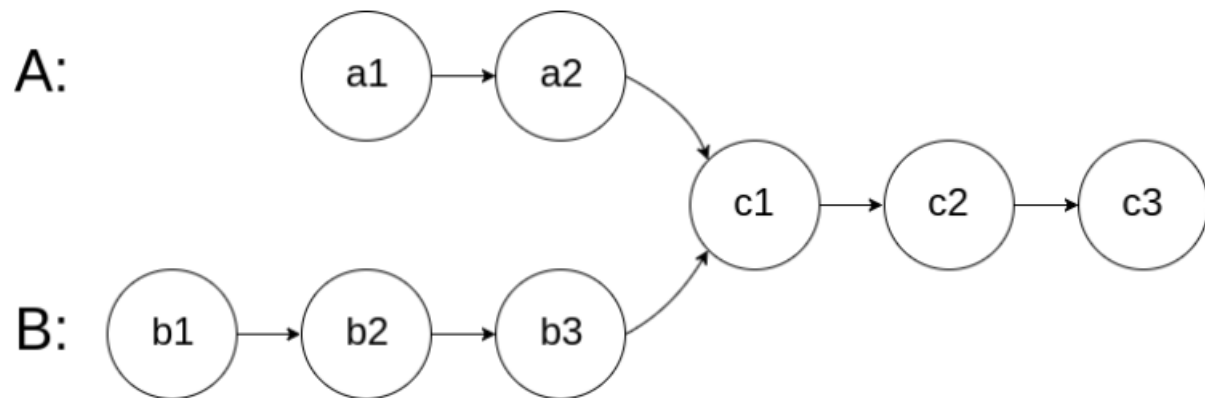
l2



160. 相交链表

编写一个程序，找到两个单链表相交的起始节点。

如下面的两个链表：



在节点 c1 开始相交。

注意：

- 如果两个链表没有交点，返回 `null`。
- 在返回结果后，两个链表仍须保持原有的结构。
- 可假定整个链表结构中没有循环。
- 程序尽量满足 $O(n)$ 时间复杂度，且仅用 $O(1)$ 内存。

■ 同样的题目

□ [面试题 02.07. 链表相交](#)

□ [面试题52. 两个链表的第一个公共节点](#)

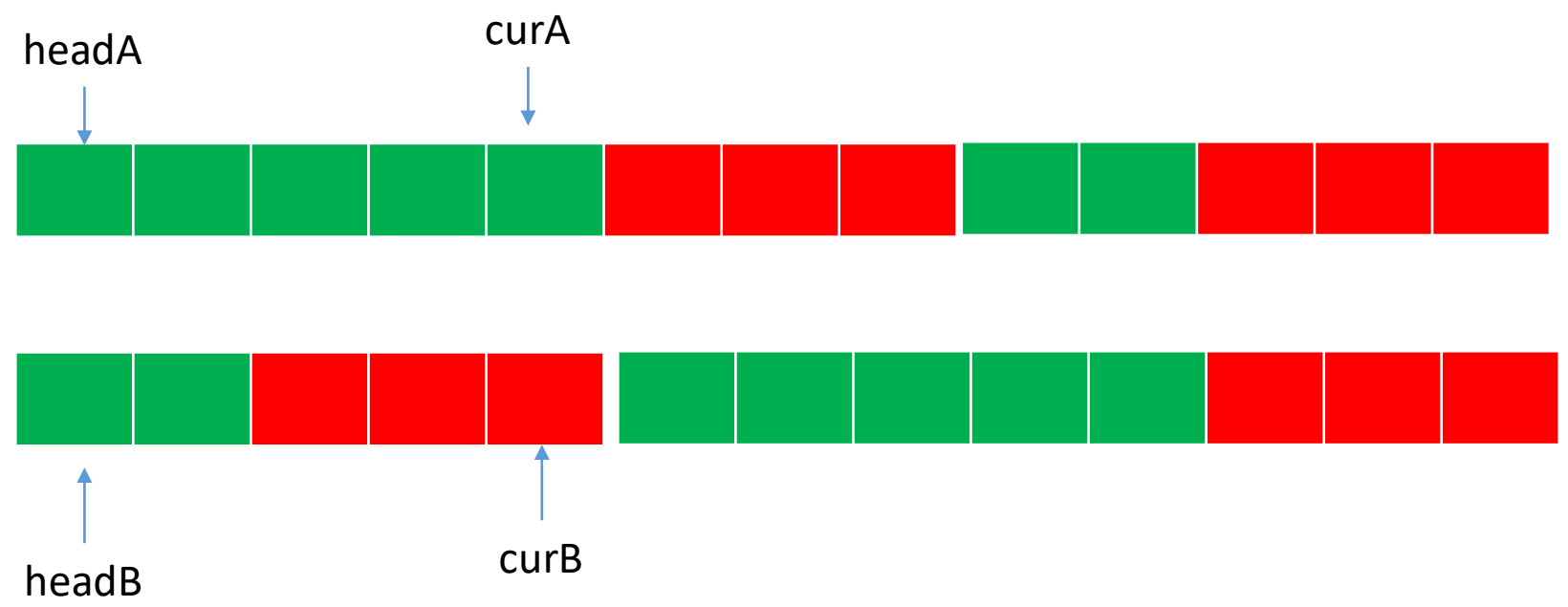
headA

curA



headB

curB



86. 分隔链表

给定一个链表和一个特定值 x ，对链表进行分隔，使得所有小于 x 的节点都在大于或等于 x 的节点之前。

你应当保留两个分区中每个节点的初始相对位置。

示例:

输入: head = 1->4->3->2->5->2, $x = 3$

输出: 1->2->2->4->3->5

■ 时间复杂度: $O(n)$ 、空间复杂度: $O(1)$

■ 如果要分隔成 3 部分呢?

□ 小于 x 的节点都在 x 左边

□ 等于 x 的节点都在中间

□ 大于 x 的节点都在 x 右边

■ 相似的题目: [面试题 02.04. 分割链表](#) (不要求保留两个分区中每个节点的初始相对位置)



234. 回文链表

请判断一个链表是否为回文链表。

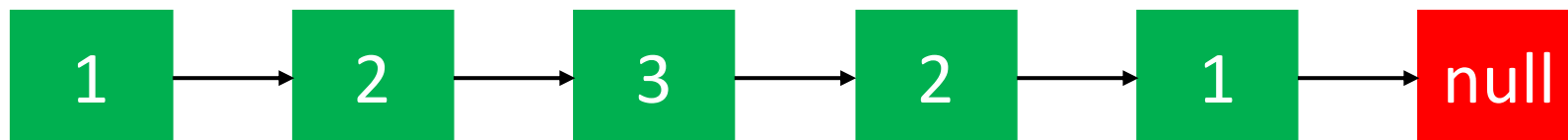
示例 1:

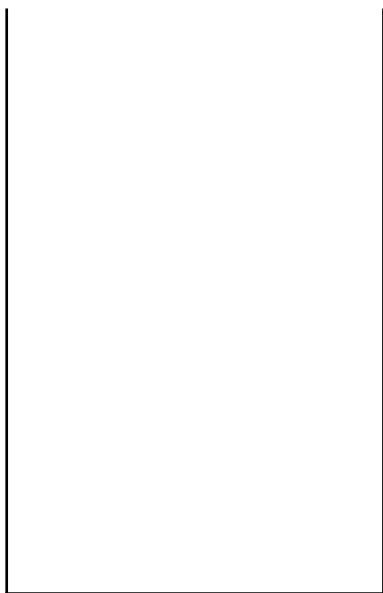
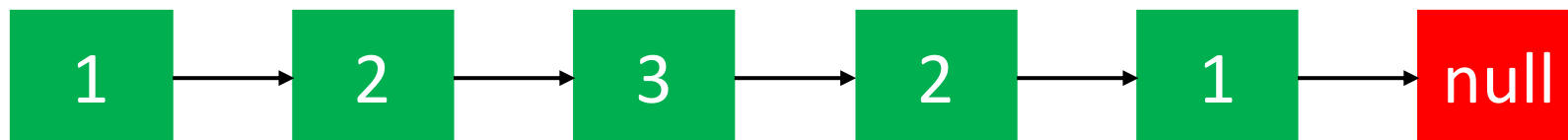
输入: 1->2
输出: false

示例 2:

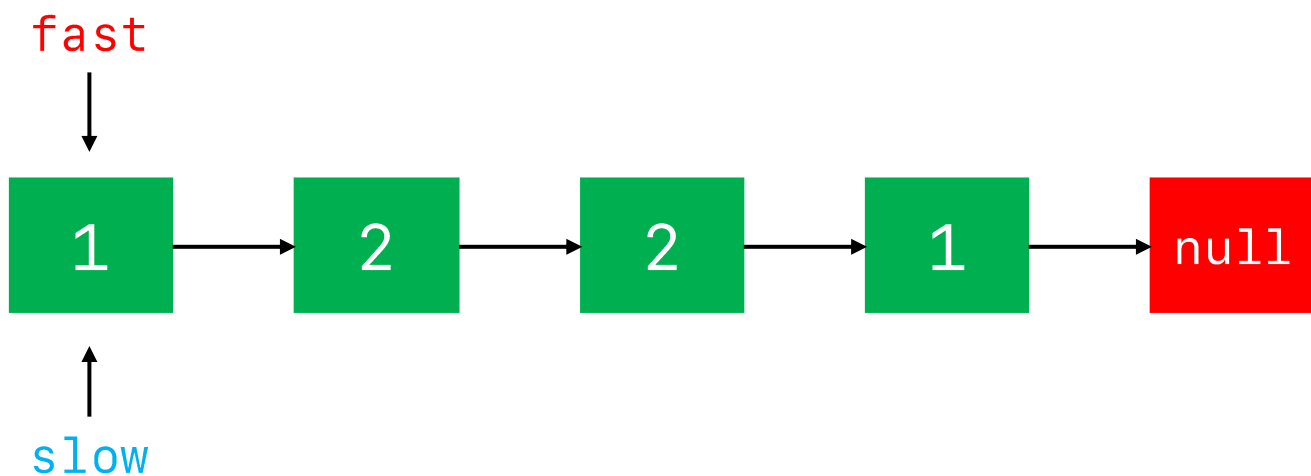
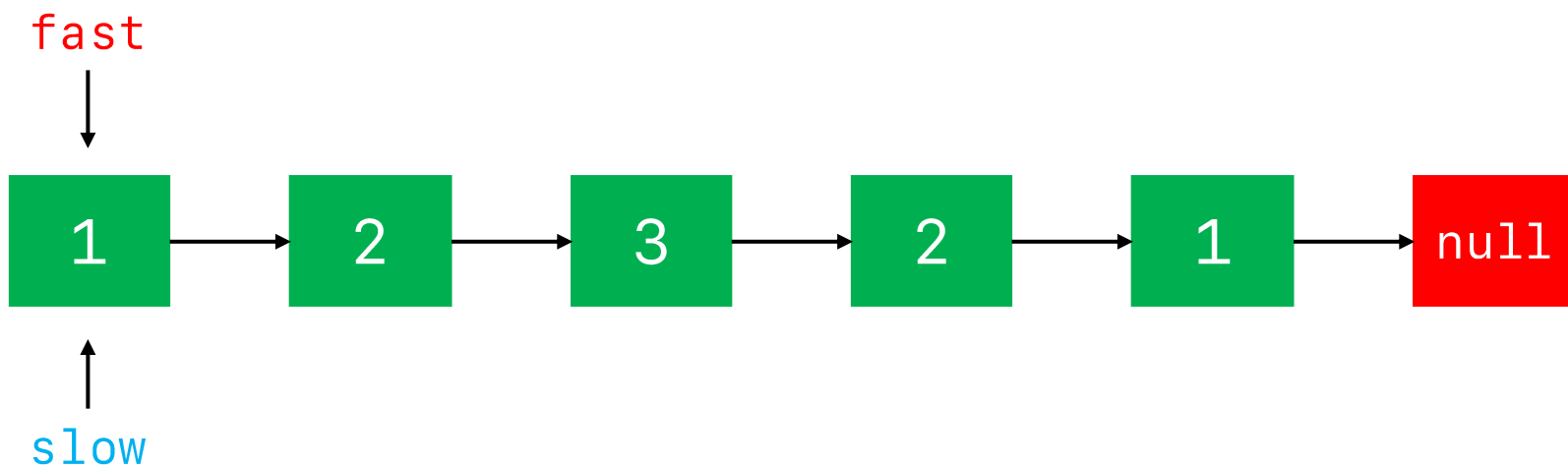
输入: 1->2->2->1
输出: true

- 时间复杂度: $O(n)$ 、空间复杂度: $O(1)$
- 同样的题目: [面试题 02.06. 回文链表](#)
- 如果要求不能破坏链表的原来结构呢?

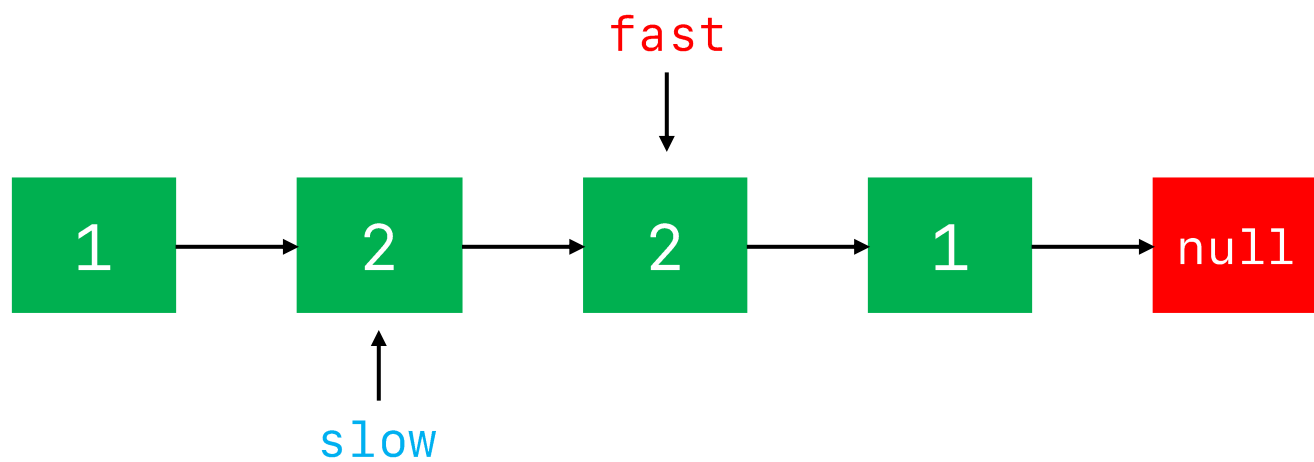
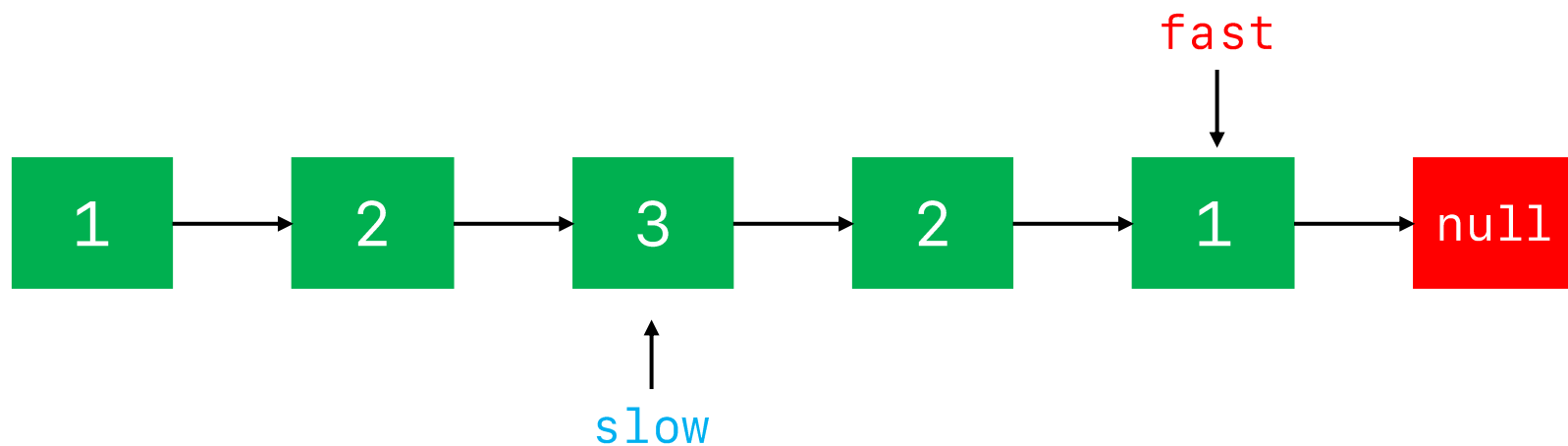




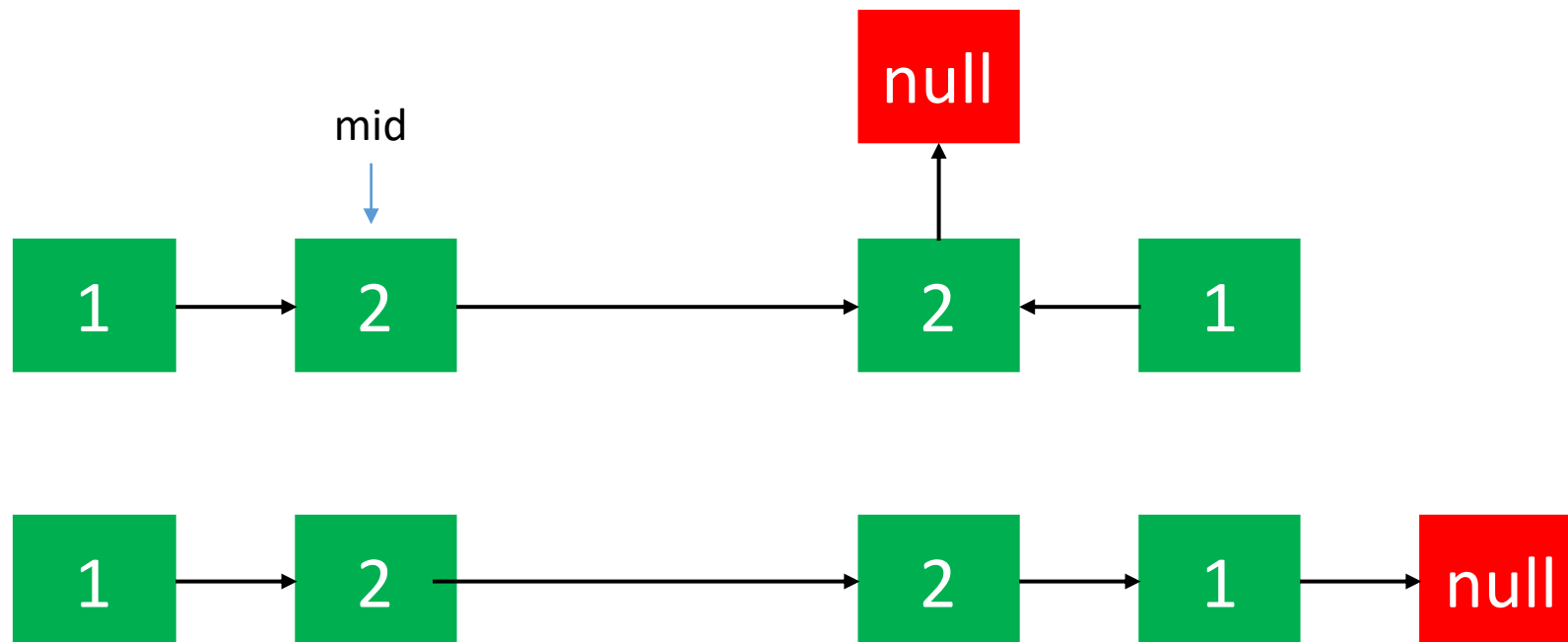
利用快慢指针得到中间节点

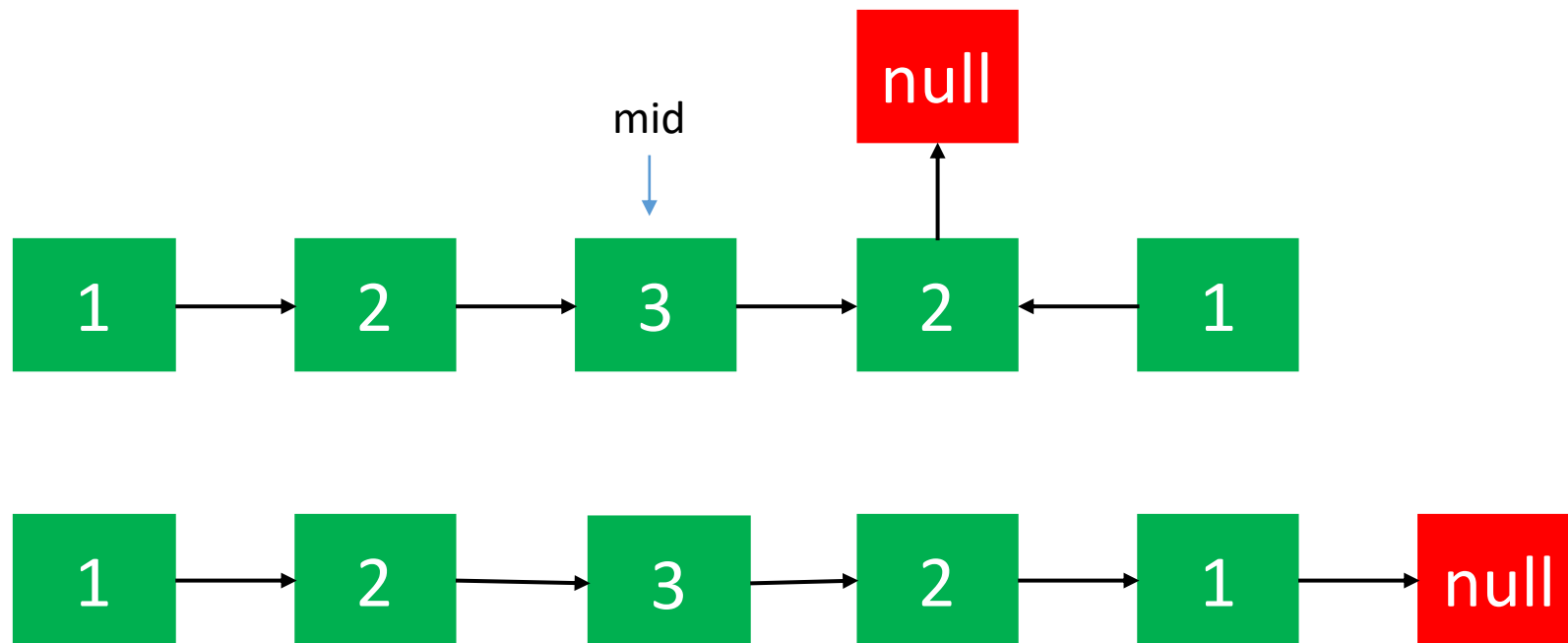


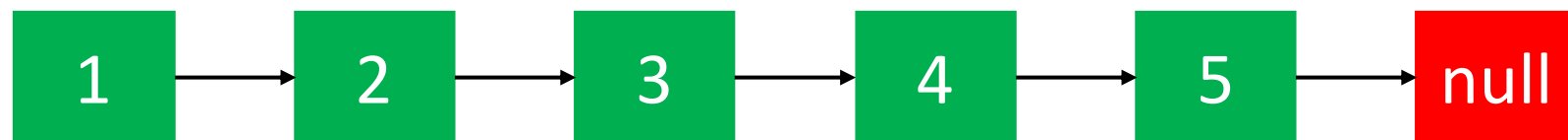
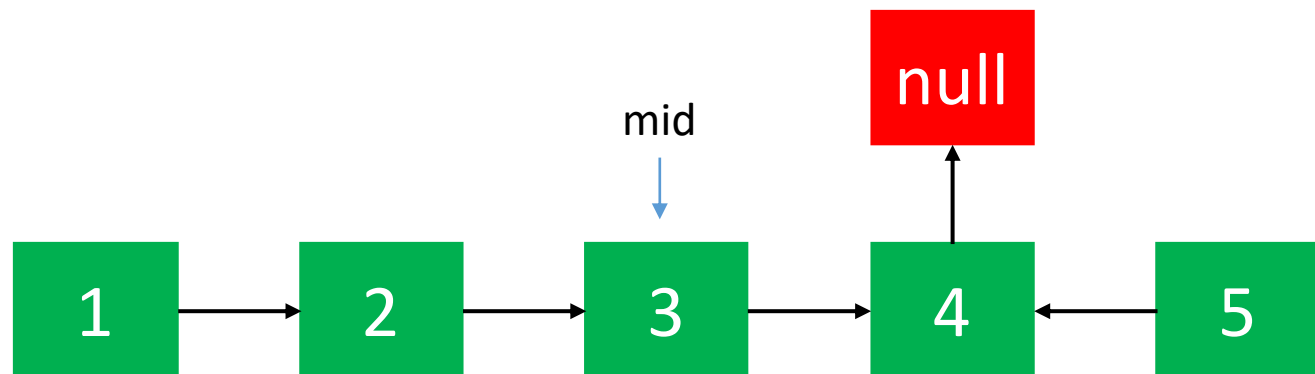
利用快慢指针得到中间节点



第1轮







思考题 — 138. 复制带随机指针的链表

给定一个链表，每个节点包含一个额外增加的随机指针，该指针可以指向链表中的任何节点或空节点。

要求返回这个链表的 **深拷贝**。

我们用一个由 `n` 个节点组成的链表来表示输入/输出中的链表。每个节点用一个 `[val, random_index]` 表示：

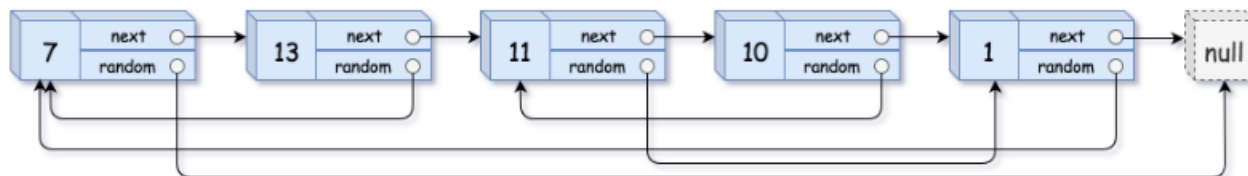
- `val`：一个表示 `Node.val` 的整数。
- `random_index`：随机指针指向的节点索引（范围从 `0` 到 `n-1`）；如果不指向任何节点，则为 `null`。

■ 时间复杂度： $O(n)$ 、空间复杂度： $O(1)$

■ 同样的题目：[面试题35. 复杂链表的复制](#)

思考题 — 138. 复制带随机指针的链表

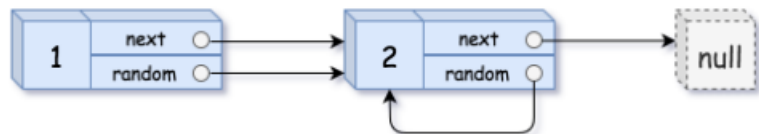
示例 1:



输入: head = [[7,null],[13,0],[11,4],[10,2],[1,0]]

输出: [[7,null],[13,0],[11,4],[10,2],[1,0]]

示例 2:

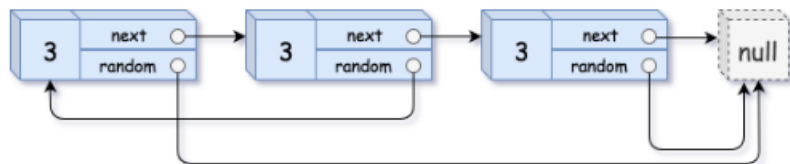


输入: head = [[1,1],[2,1]]

输出: [[1,1],[2,1]]

思考题 — 138. 复制带随机指针的链表

示例 3:



输入: head = [[3,null],[3,0],[3,null]]

输出: [[3,null],[3,0],[3,null]]

示例 4:

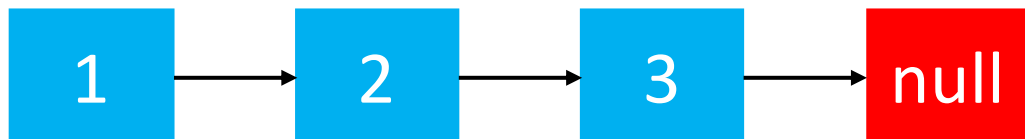
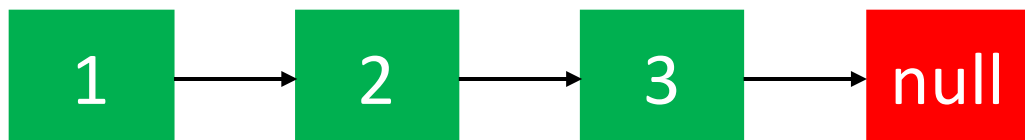
输入: head = []

输出: []

解释: 给定的链表为空（空指针），因此返回 null。

提示:

- `-10000 <= Node.val <= 10000`
- `Node.random` 为空 (null) 或指向链表中的节点。
- 节点数目不超过 1000。



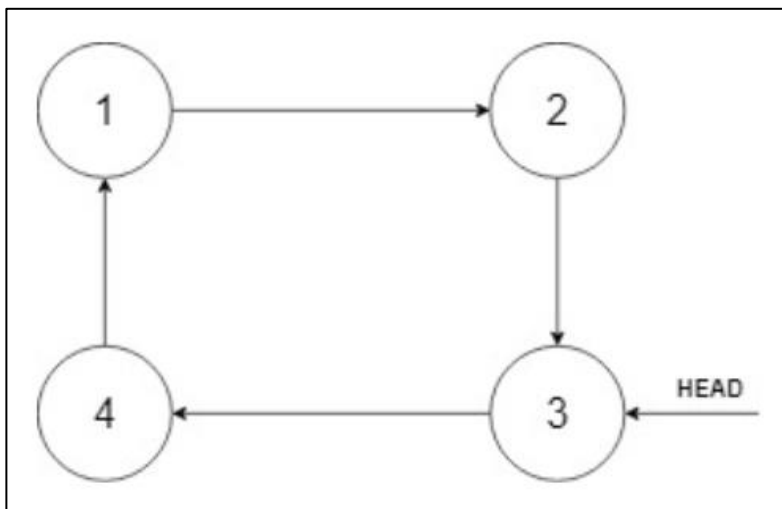
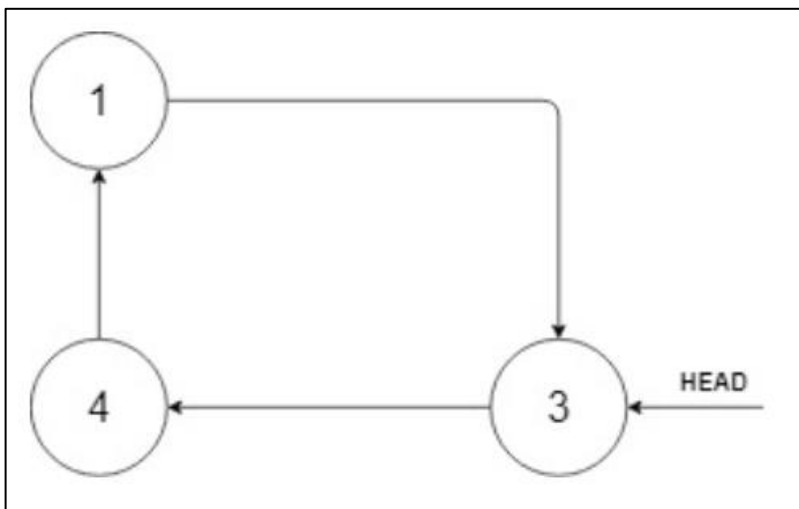
思考题 — 708. 循环有序列表的插入

给定循环升序列表中的一个点，写一个函数向这个列表中插入一个新元素，使这个列表仍然是循环升序的。给定的可以是这个列表中任意一个顶点的指针，并不一定是这个列表中最小元素的指针。

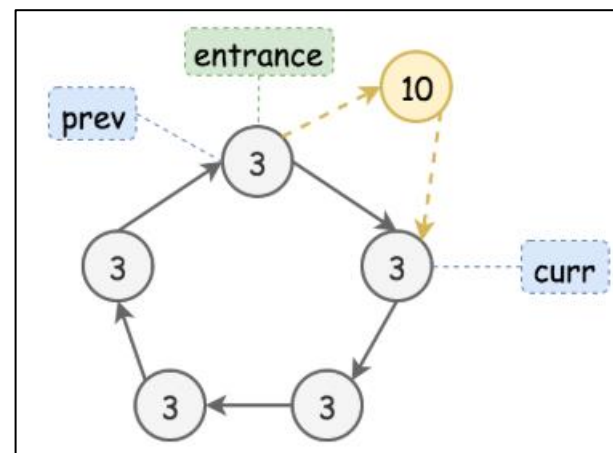
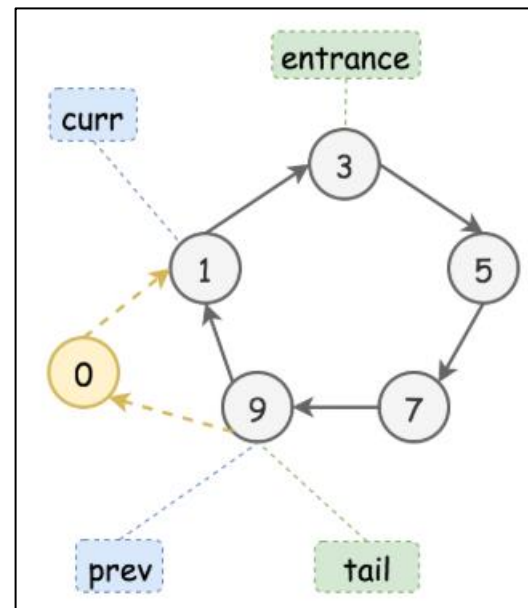
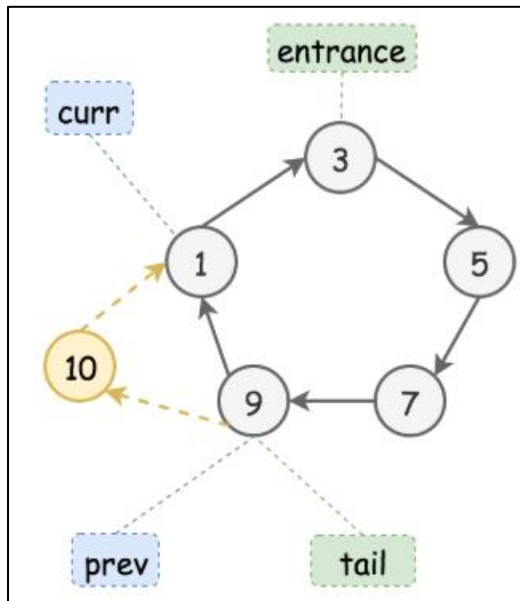
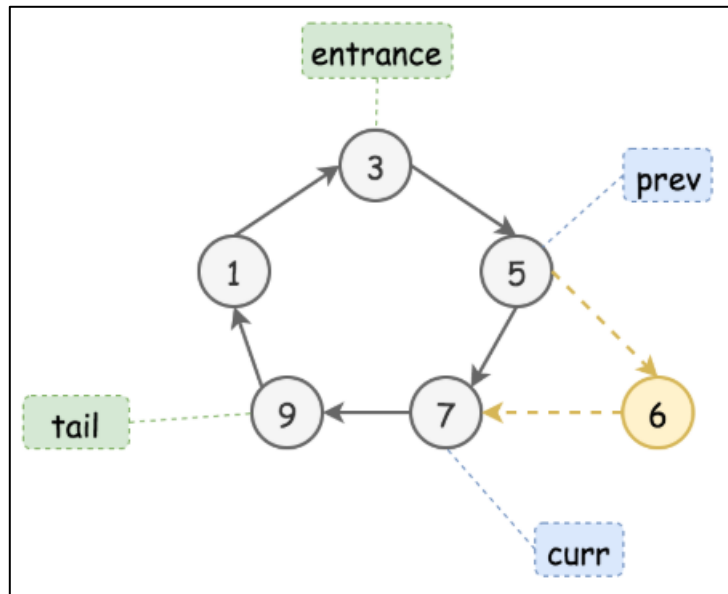
如果有多个满足条件的插入位置，你可以选择任意一个位置插入新的值，插入后整个列表仍然保持有序。

如果列表为空（给定的节点是 `null`），你需要创建一个循环有序列表并返回这个点。否则，请返回原先给定的节点。

■ 时间复杂度： $O(n)$ 、空间复杂度： $O(1)$



思考题 — 708. 循环有序列表的插入



思考题 — 25. K个一组翻转链表

给你一个链表，每 k 个节点一组进行翻转，请你返回翻转后的链表。

k 是一个正整数，它的值小于或等于链表的长度。

如果节点总数不是 k 的整数倍，那么请将最后剩余的节点保持原有顺序。

示例：

给定这个链表：1→2→3→4→5

当 $k = 2$ 时，应当返回：2→1→4→3→5

当 $k = 3$ 时，应当返回：3→2→1→4→5

说明：

- 你的算法只能使用常数的额外空间。
- **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。

■ 时间复杂度： $O(n)$ 、空间复杂度： $O(1)$

■ 相似的题目：[24. 两两交换链表中的节点](#)

- [237. 删除链表中的节点](#) ([第一季](#)中讲过)
- [141. 环形链表](#) ([第一季](#)中讲过)
- [206. 反转链表](#)、[面试题24. 反转链表](#) ([第一季](#)中讲过)
- [21. 合并两个有序链表](#) ([每周一到算法题](#)中讲过)
- [23. 合并K个排序链表](#) ([每周一到算法题](#)中讲过)