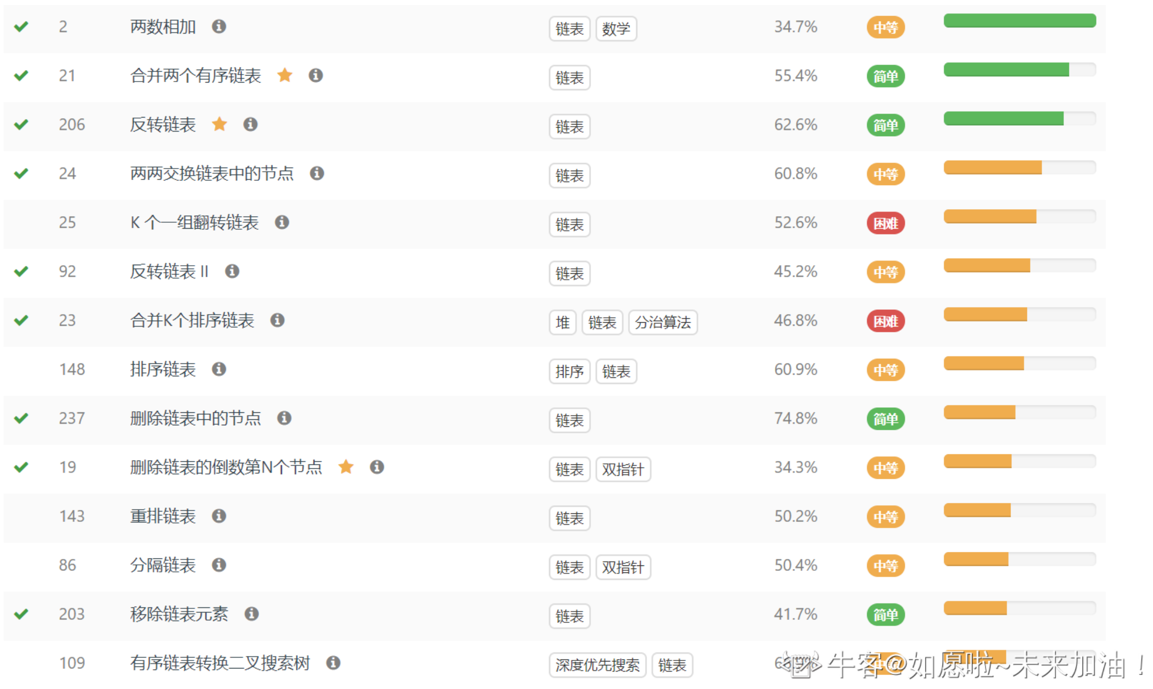
@王先林@吴东箭

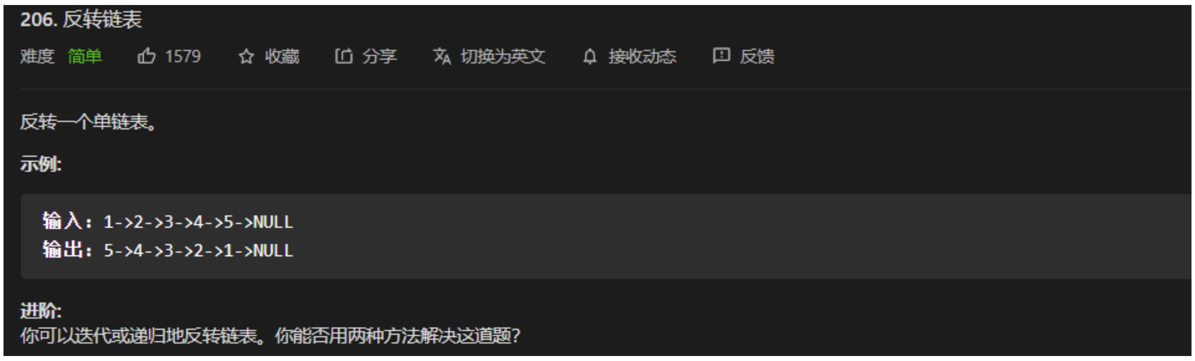
@日拱一卒，功不唐捐

@We are all in the gutter, but some of usare looking at the stars.

linked list



✓	2	两数相加	链表 数学	34.7%	中等
✓	21	合并两个有序链表 ★	链表	55.4%	简单
✓	206	反转链表 ★	链表	62.6%	简单
✓	24	两两交换链表中的节点	链表	60.8%	中等
	25	K 个一组翻转链表	链表	52.6%	困难
✓	92	反转链表 II	链表	45.2%	中等
✓	23	合并K个排序链表	堆 链表 分治算法	46.8%	困难
	148	排序链表	排序 链表	60.9%	中等
✓	237	删除链表中的节点	链表	74.8%	简单
✓	19	删除链表的倒数第N个节点 ★	链表 双指针	34.3%	中等
	143	重排链表	链表	50.2%	中等
	86	分隔链表	链表 双指针	50.4%	中等
✓	203	移除链表元素	链表	41.7%	简单
	109	有序链表转换二叉搜索树	深度优先搜索 链表	[illegible]	中等



206. 反转链表

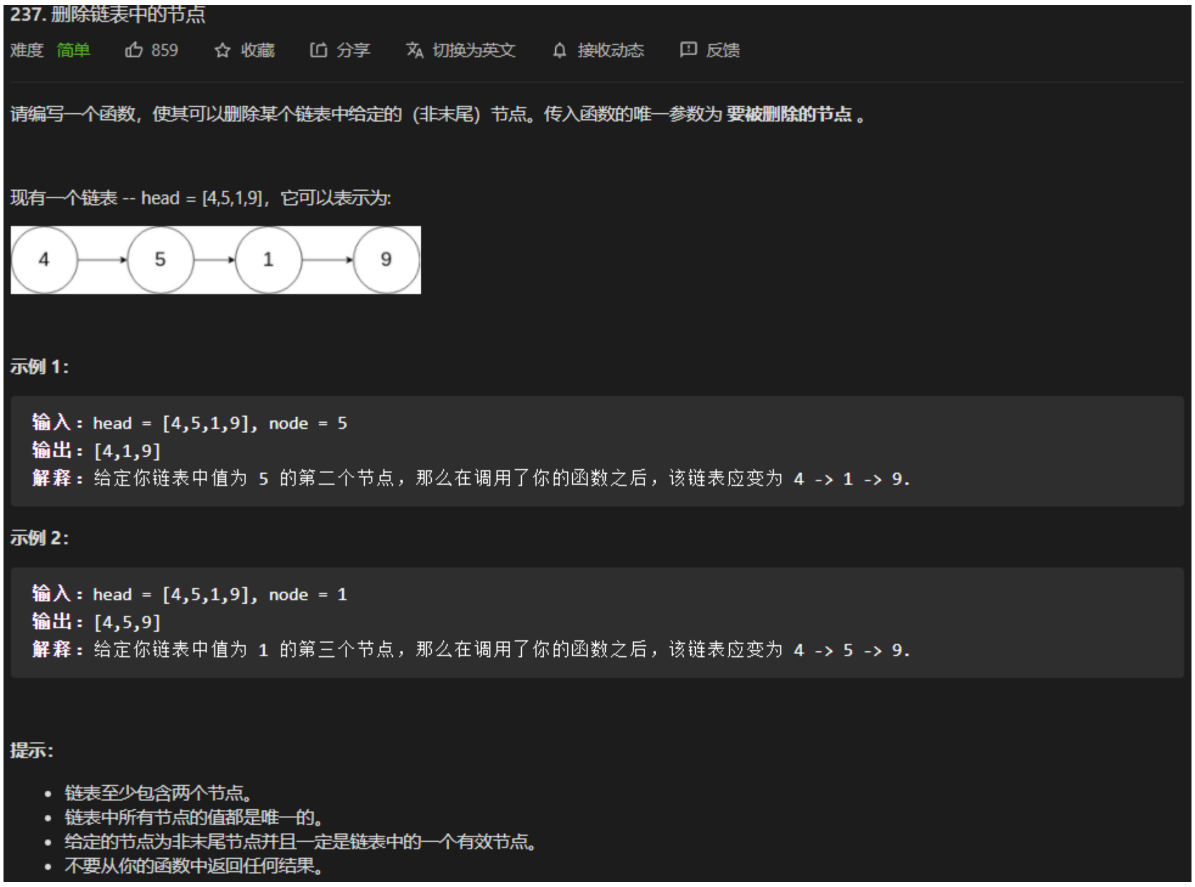
反转一个单链表。

示例:

```
输入: 1->2->3->4->5->NULL
输出: 5->4->3->2->1->NULL
```

进阶:
你可以迭代或递归地反转链表。你能否用两种方法解决这道题?

```
class Solution {
    public ListNode reverseList(ListNode head) {
        if (head == null || head.next == null) return head;
        ListNode s = head;
        ListNode f = head.next;
        while (f != null && f.next != null) {
            ListNode mid = f.next;
            f.next = s;//
            s = f;
            f = mid;
        }
        //处理最后一个结点
        head.next = null;   //Error - Found cycle in the ListNode,如果不处理的话
        f.next = s;
        return f;
    }
}
```

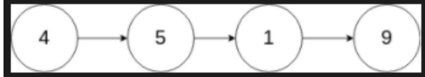


237. 删除链表中的节点

难度 简单 859 收藏 分享 切换为英文 接收动态 反馈

请编写一个函数，使其可以删除某个链表中给定的（非末尾）节点。传入函数的唯一参数为 **要被删除的节点** 。

现有一个链表 -- head = [4,5,1,9]，它可以表示为:



示例 1:

```
输入：head = [4,5,1,9], node = 5
输出：[4,1,9]
解释：给定你链表中值为 5 的第二个节点，那么在调用了你的函数之后，该链表应变为 4 -> 1 -> 9.
```

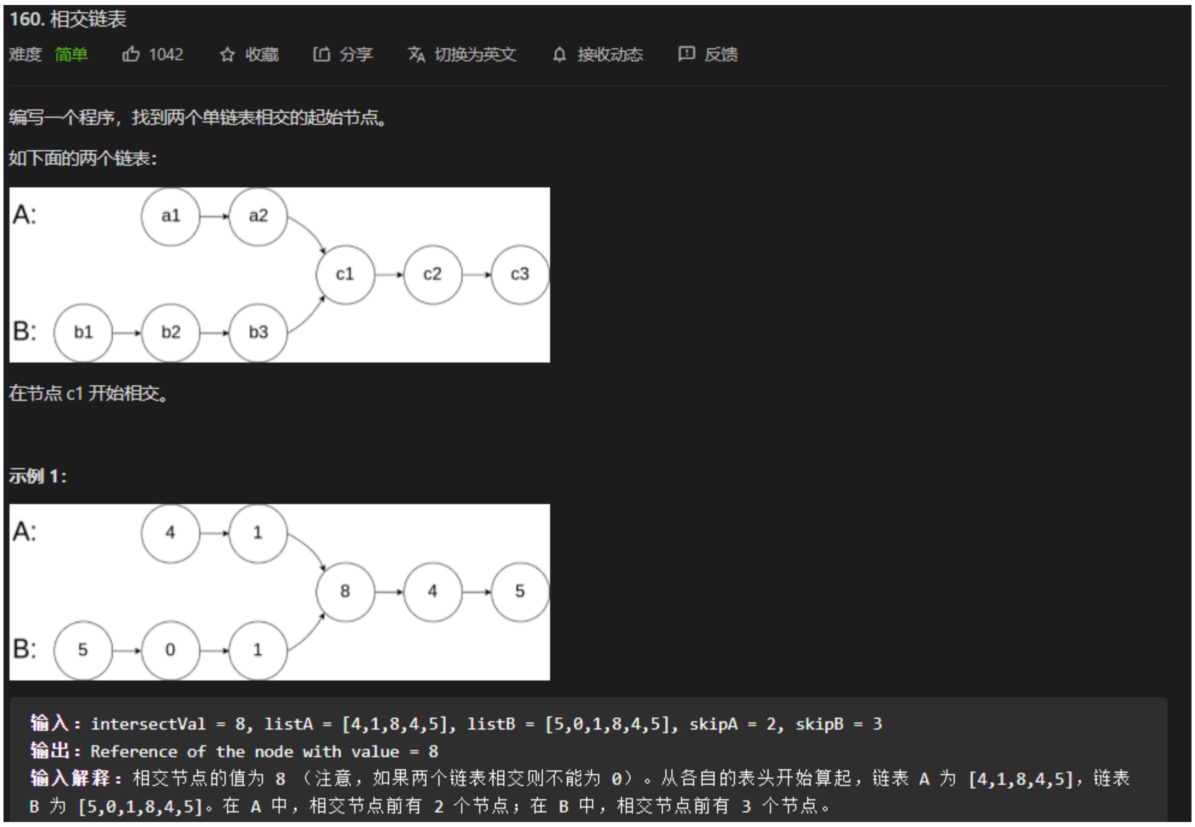
示例 2:

```
输入：head = [4,5,1,9], node = 1
输出：[4,5,9]
解释：给定你链表中值为 1 的第三个节点，那么在调用了你的函数之后，该链表应变为 4 -> 5 -> 9.
```

提示:

- 链表至少包含两个节点。
- 链表中所有节点的值都是唯一的。
- 给定的节点为非末尾节点并且一定是链表中的一个有效节点。
- 不要从你的函数中返回任何结果。

```
/**
 * Definition for singly-linked list.
 * public class ListNode {
 *     int val;
 *     ListNode next;
 *     ListNode(int x) { val = x; }
 * }
 */
class Solution {
    public void deleteNode(ListNode node) {
        ListNode mid = node;
        ListNode slow = node;
        while (mid.next != null) {
            mid.val = mid.next.val;
            slow = mid;
            mid = mid.next;
        }
        //如何删除最后一个结点？    再用slow指针记录前结点，让其指向null
        slow.next = null;

    }
}
```

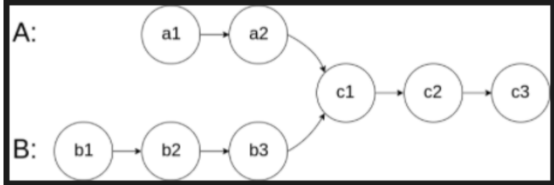


160. 相交链表

难度 简单 1042 收藏 分享 切换为英文 接收动态 反馈

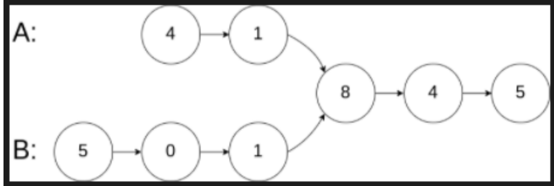
编写一个程序，找到两个单链表相交的起始节点。

如下面的两个链表:



在节点 c1 开始相交。

示例 1：

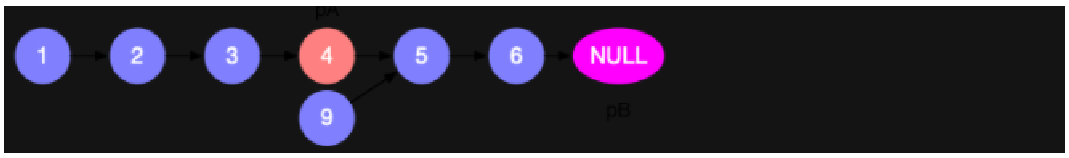


```
输入：intersectVal = 8, listA = [4,1,8,4,5], listB = [5,0,1,8,4,5], skipA = 2, skipB = 3
输出：Reference of the node with value = 8
输入解释：相交节点的值为 8 （注意，如果两个链表相交则不能为 0）。从各自的表头开始算起，链表 A 为 [4,1,8,4,5]，链表
B 为 [5,0,1,8,4,5]。在 A 中，相交节点前有 2 个节点；在 B 中，相交节点前有 3 个节点。
```

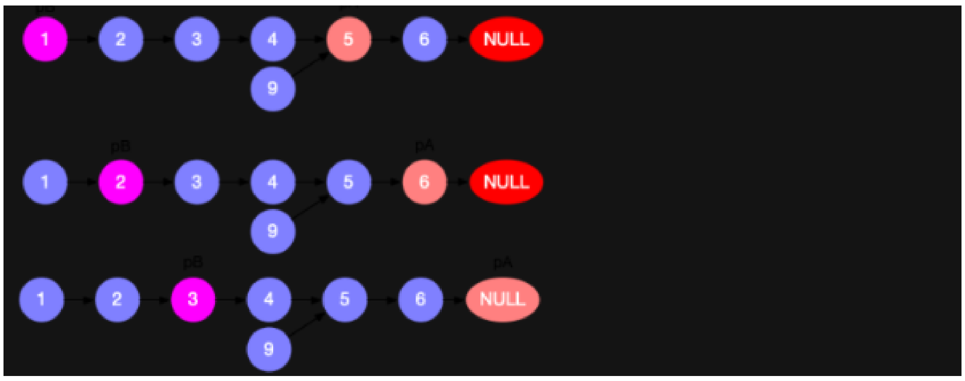
```
class ListNode {
    int val;
    ListNode next;

    ListNode(int x) {
        val = x;
        next = null;
    }
}

/*
```

```
文艺版解释：听闻远方有你，动身跋涉千里，我吹过你吹过的风，这算不算相拥？我走过你走过的路，这算不算相逢？
走到尽头见不到你，于是走过你来时的路，等到相遇时才发现，你也走过我来时的路

浪漫的算法
2021/3/17
11:59
*/

class Solution {
    public ListNode getIntersectionNode(ListNode headA, ListNode headB) {

        while (headA == null || headB == null) return null;


        ListNode pa = headA;
        ListNode pb = headB;
        while (pa != pb) {
            pa = pa == null ? headB : pa.next;
            pb = pb == null ? headA : pb.next;
        }
        return pa;
    }
}
```

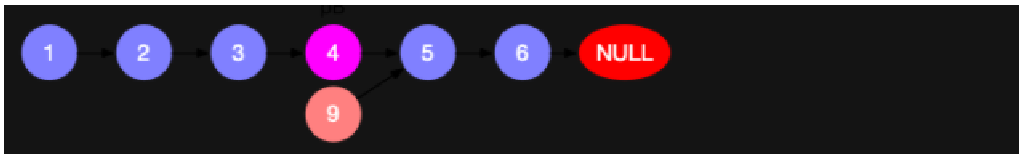
//走到尽头见不到你



//于是走过你来时的路



//等到相遇时才发现



下面是实现后