**难度: 简单**

输入两个链表，找出它们的第一个公共节点。

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，如果两个列表相交则不能为 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。

示例 2：



输入：intersectVal = 2, listA = [0,9,1,2,4], listB = [3,2,4], skipA = 3, skipB = 1

输出：Reference of the node with value = 2

输入解释：相交节点的值为 2 （注意，如果两个列表相交则不能为 0）。从各自的表头开始算起，链表 A 为 [0,9,1,2,4]，链表 B 为 [3,2,4]。在 A 中，相交节点前有 3 个节点；在 B 中，相交节点前有 1 个节点。

示例 3：



输入：intersectVal = 0, listA = [2,6,4], listB = [1,5], skipA = 3, skipB = 2

输出：null

输入解释：从各自的表头开始算起，链表 A 为 [2,6,4]，链表 B 为 [1,5]。由于这两个链表不相交，所以 intersectVal 必须为 0，而 skipA 和 skipB 可以是任意值。

解释：这两个链表不相交，因此返回 null。

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。

本题与主站 160 题相同：https://leetcode-cn.com/problems/intersection-of-two-linked-lists/

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/liang-ge-lian-biao-de-di-yi-ge-gong-gong-jie-dian-lcof

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

**思路1:**

1). 先判断两个节点是否为空,如果其中有一个为空,直接返回null

2). 如果1不成立,把两个节点,分别给到两个临时节点

3). 循环临时节点1 不等于 临时节点2

4). 当临时节点1为空的时候,把headB给到它

5). 当临时节点2为空的时候,把headA给到它

6). 最后可以返回临时节点1或者临时节点2

**注:** 不用担心死循环,因为它们两个节点会得到null,就会结束循环

|  |
| --- |
| public ListNode getIntersectionNode(ListNode headA, ListNode headB) {  if (headA == null || headB == null) {  return null;  }  ListNode temp1 = headA;  ListNode temp2 = headB;  while(temp1 != temp2) {  temp1 = temp1 == null ? headB : temp1.next;  temp2 = temp2 == null ? headA : temp2.next;  }  return temp1;  } |