# 题目

给定两个单链表的头节点 headA 和 headB ，请找出并返回两个单链表相交的起始节点。如果两个链表没有交点，返回 null 。

图示两个链表在节点 c1 开始相交：



题目数据 保证 整个链式结构中不存在环。

注意，函数返回结果后，链表必须 保持其原始结构 。

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：Intersected at '8'

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。

示例 2：



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

输出：Intersected at '2'

解释：相交节点的值为 2 （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 [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

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。

这两个链表不相交，因此返回 null。

**提示：**

listA 中节点数目为 m

listB 中节点数目为 n

0 <= m, n <= 3 \* 104

1 <= Node.val <= 105

0 <= skipA <= m

0 <= skipB <= n

如果 listA 和 listB 没有交点，intersectVal 为 0

如果 listA 和 listB 有交点，intersectVal == listA[skipA + 1] == listB[skipB + 1]

进阶：能否设计一个时间复杂度 O(n) 、仅用 O(1) 内存的解决方案？

注意：本题与主站 160 题相同：

https://leetcode-cn.com/problems/intersection-of-two-linked-lists/

# 分析