

面试题0207-链表相交

题述

面试题 02.07. 链表相交

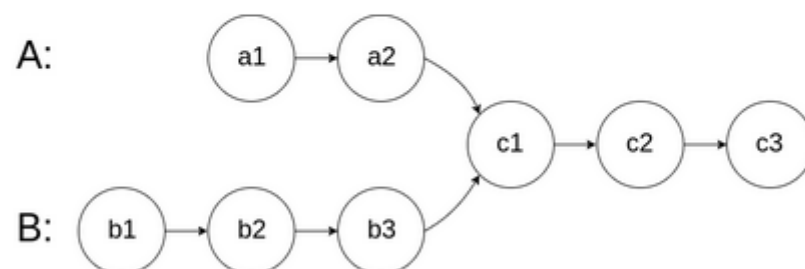
难度 简单

131



给你两个单链表的头节点 `headA` 和 `headB`，请你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`。

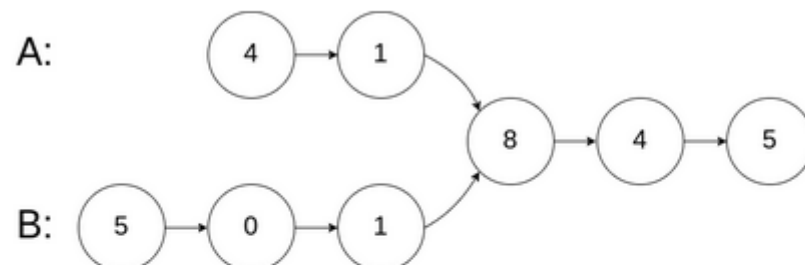
图示两个链表在节点 `c1` 开始相交：



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）。

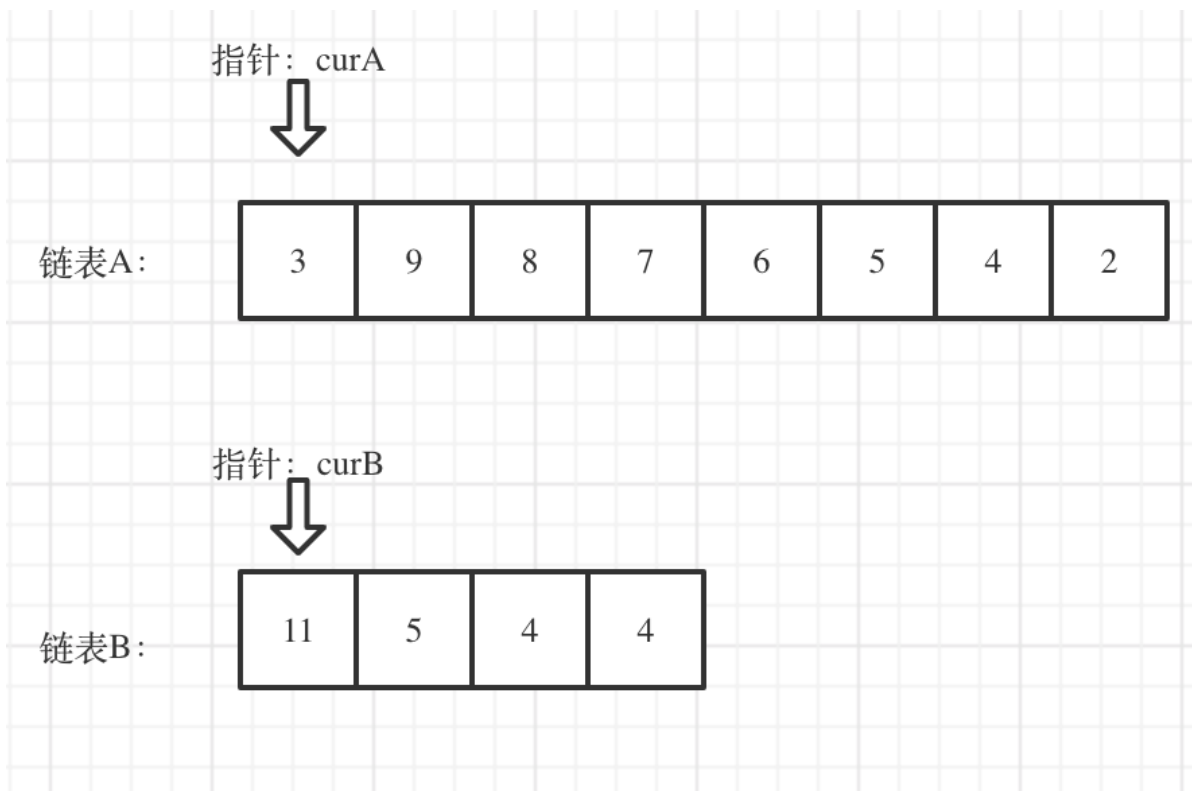
从各自的表头开始算起，链表 A 为 [4,1,8,4,5]，链表 B 为 [5,0,1,8,4,5]。

在 A 中，相交节点前有 2 个节点；在 B 中，相交节点前有 3 个

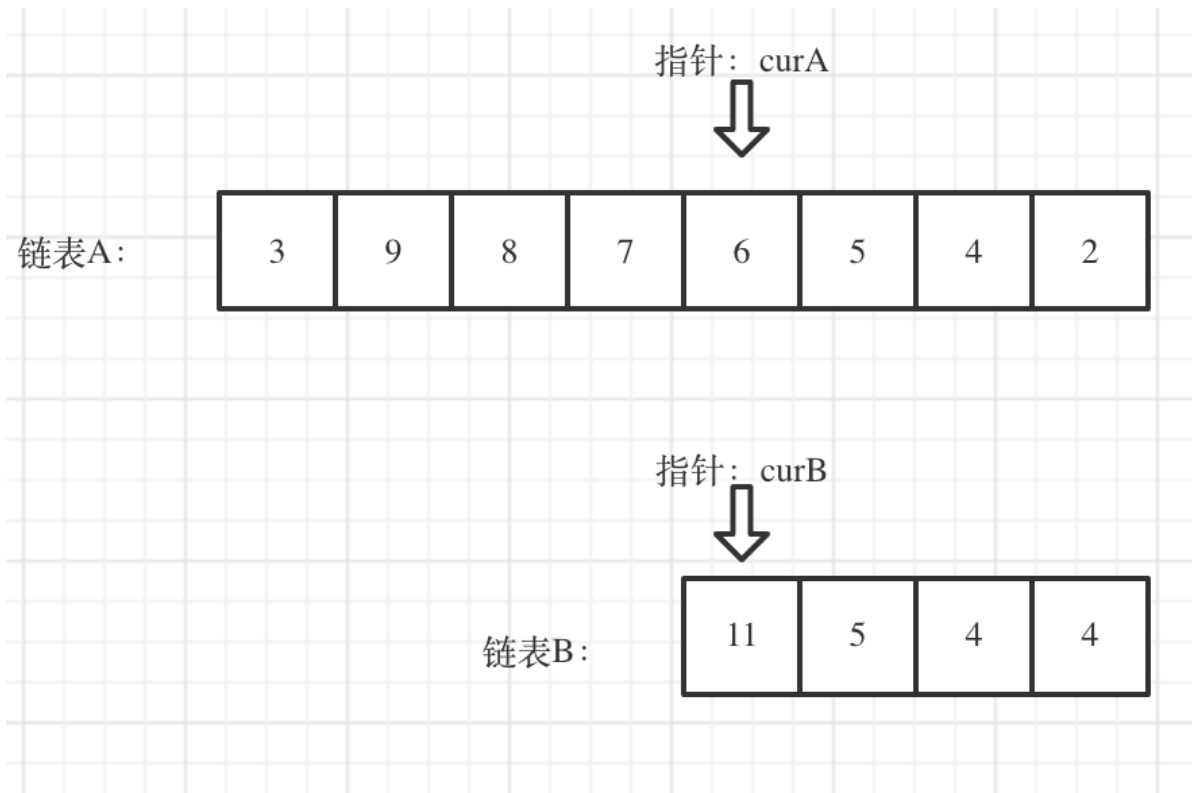
思路

这道题的关键在于，一定要注意，所谓的交点指的并不是简单的数值相同，而是指针相同，即地址相同。

看如下两个链表，目前curA指向链表A的头结点，curB指向链表B的头结点：



我们求出两个链表的长度，并求出两个链表长度的差值，然后让curA移动到，和curB 末尾对齐的位置，如图：



此时我们就可以比较curA和curB是否相同，如果不相同，同时向后移动curA和curB，如果遇到curA == curB，则找到交点。

否则循环退出返回空指针。

题解

```
/**  
 * Definition for singly-linked list.
```

```

* struct ListNode {
*     int val;
*     ListNode *next;
*     ListNode(int x) : val(x), next(NULL) {}
* };
*/
class Solution {
public:
    ListNode *getIntersectionNode(ListNode *headA, ListNode *headB)
    {
        //相交指的不仅仅是简单的数值相等
        //而是指针地址相等 本质上的完全相等

        ListNode* curA=headA;
        ListNode* curB=headB;
        int lengthA=0,lengthB=0;
        while(curA!=nullptr)
        {
            //求链表A的长度
            lengthA++;
            curA=curA->next;
        }

        while(curB!=nullptr)
        {
            //求链表B的长度
            lengthB++;
            curB=curB->next;
        }

        curA=headA;
        curB=headB;

        //让curA成为最长链表的头 lengthA为其长度
        if(lengthB>lengthA)
        {
            //实施交换
            swap(lengthA,lengthB);
            swap(curA,curB);
        }

        //求长度差
        int gap=lengthA-lengthB;

        //让curA和curB处于同一起点上（末尾位置对齐）
        while(gap-->0)
        {
            curA=curA->next;
        }

        //遍历curA和curB 遇到相同就直接返回
        while(curA!=nullptr)
        {
            if(curA==curB)
            {
                return curA;
            }
            curA=curA->next;
        }
    }
};

```

```
        curB=curB->next;
    }
    return nullptr;
}
};
```

执行结果: **通过** [显示详情 >](#)

[添加备注](#)

执行用时: **40 ms** , 在所有 C++ 提交中击败了 **66.09%** 的用户

内存消耗: **14.2 MB** , 在所有 C++ 提交中击败了 **67.90%** 的用户

通过测试用例: **45 / 45**

炫耀一下:



[写题解, 分享我的解题思路](#)

提交结果	执行用时	内存消耗	语言	提交时间	备注
通过	40 ms	14.2 MB	C++	2021/12/28 16:00	添加备注
解答错误	N/A	N/A	C++	2021/12/28 15:57	添加备注
通过	40 ms	14.2 MB	C++	2021/12/28 15:57	添加备注
解答错误	N/A	N/A	C++	2021/12/28 15:56	添加备注

总结

一开始在写判断条件时 漏了一个=号 导致错误

注意细节