

剑指52 两个链表的第一个公共节点

链表类题目。没想出来。重要~~

掌握思路：

双指针操作技巧 或 快慢指针技巧

头尾衔接的思考方式

标准参考题解（你的名字，浪漫相遇）：

1. 因为题目要求 $O(n)$ 的时间复杂度，不能嵌套循环暴力解题；因为要求 $O(1)$ 的空间复杂度，所以不能使用哈希表进行哈希比较来解题；
2. 所以剩下的解题方向只有进行单次的迭代遍历，则解题的思路重点就放在了，如果链表有交点，如何保障每次做单个节点向前迭代时，两个链表如何能正好同时迭代到相交的节点；
 - a. 假设链表headA的长度为a，链表headB的长度为b，两个链表相交后重合的节点长度为c；若两条链表不相交，则 $c=0$ ；
 - b. 如果两个链表有交点，如果要限制两个链表能刚好同时遍历到相交的节点，则要使得两个链表的遍历路径长度要相同(len)，即两个链表遍历到交点时，总步进节点数都为 $(len - c)$ ；
 - c. 问题就转化为：如何能够使得两个链表的遍历长度相同的问题，此时可以想到：使得两个链表节点遍历的长度 $len=a+b$ ，即当链表指针遍历到单个链表的尾部时，直接转移遍历指针移动到另一条链表的头部继续进行遍历，如果碰到相同节点，直接返回，直到两个遍历指针同时指向了链表的尾部为结束
 - d. 实现时注意边界的处理；

（即2个链表同时这样做：当前链表遍历完后接上另一个链表遍历。这样在第一次遍历另一个链表的过程中，2个指针必相遇在2个链表的第1公共节点。）

（边界情况注意点：把2个链表末尾的NULL也加入遍历，这样才能考虑到链表无相交的情况， $a+b = b+a$ 时，2方正好都遍历到了NULL说明没有交点，退出循环。）

```
1 // 标准
2 class Solution {
3 public:
4     ListNode *getIntersectionNode(ListNode *headA, ListNode *headB) {
5         ListNode* A = headA;
6         ListNode* B = headB;
7         while (A != B) {
8             if (A != NULL)
9                 A = A->next;
10            else
11                A = headB;
12
13            if (B != NULL)
14                B = B->next;
15            else
16                B = headA;
17        }
18        return A;
19    }
```

```
19     }  
20 };  
21
```

其他方案一：差值法

如果两个链表有公共部分，则为了让其两个指针同时移动并能相遇，因此需要调整起点位置。即较长链表的遍历起始位置不在开头位置，而是向后移动 $|a-b|$ 个节点时，再开始遍历。这样，双方能同时到达结尾，也意味着双方能同时到达第一个公共点。

其他方案二：hash表（稍微比 n^2 暴力循环好一点）

使用set数据结构，插入所有的headA链表结点地址。

遍历headB链表，每次访问一个结点就去哈希表中查找是否已经出现过了。

其他方案三：hash表（稍微比 n^2 暴力循环好一点）

分别使用两个栈存储每个链表的结点

然后同时弹出两个栈的结点，如果栈顶的结点相同，则需要继续弹出，直到不相同为止。