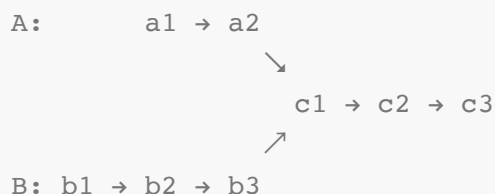


## 160. Intersection of Two Linked Lists

**题目描述：** <https://leetcode.com/problems/intersection-of-two-linked-lists/>

找到两个链表相同的部分，如果不存在返回null，在这里相同的意思是从哪个地方开始到结尾完全一样，则返回开始的节点。

例如：

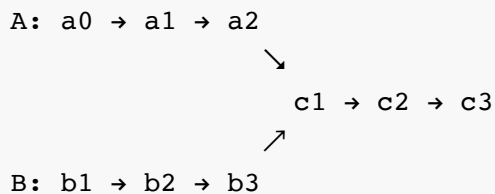


则两个链表A,B从c1处开始相交。

### 解题思路：

如果两个链表有相交的部分，则把两个链表补做长度相同，且要查询的都在后半段，然后从各自的head开始查询 一定能找到相交部分。

例如：上图的A和B，我们先把它补齐成为：



然后从头开始查就可以找到相交部分了，但是如何补齐呢？

最简单的方法就是把A补成BA，把B补成AB

这样长度相同，肯定可以拿到相交部分的结果。

### 代码：

```
/**
 * 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) {
        ListNode * cur1 = headA, *cur2 = headB;
        while(cur1!=cur2){
            cur1 = cur1?cur1->next:headB;
            cur2 = cur2?cur2->next:headA;
        }
        return cur1;
    }
};
```