一、题目说明

题目160. Intersection of Two Linked Lists，计算两个链表相连的位置。难度是Easy!

二、我的解答

这个题目，简单思考一下还是容易的。一次遍历，找到`ListA、ListB`的最后一个元素及其长度，如果`endA==endB`则相交。先移动长链表的指针`abs(numA-numB)`，然后找到相等的位置即可。

代码如下：

```
class Solution{
    public:
        ListNode* getIntersectionNode(ListNode* headA, ListNode* headB){
            if(headA==NULL || headB==NULL) return NULL;
            ListNode* endA = headA,*endB=headB;
            int numA = 1,numB = 1;

            //find the last element of headA
            while(endA->next !=NULL){
                endA = endA->next;
                numA++;
            }

            //find the last element of headB
            while(endB->next !=NULL){
                endB = endB->next;
                numB++;
            }
            if(endA != endB){
                return NULL;
            }else{
                endA = headA;
                endB = headB;
                if(numA>numB){
                    int t = numA-numB;
                    while(t>0){
                        endA = endA->next;
                        t--;
                    }
                }else if(numA<numB){
                    int t = numB-numA;
                    while(t>0){
                        endB = endB->next;
                        t--;
                    }
                }

                while(endA != endB){
                    endA = endA->next;
                    endB = endB->next;
                }
                return endA;
            }
        }
};
```

性能如下：

```
Runtime: 44 ms, faster than 96.04% of C++ online submissions for Intersection of Two Linked Lists.
Memory Usage: 16.9 MB, less than 59.26% of C++ online submissions for Intersection of Two Linked Lists.
```

三、优化措施

可以继续优化，但意义不大。