160 Intersection of Two Linked Lists

2018年4月3日 15:27

Question:

Write a program to find the node at which the intersection of two singly linked lists begins.

For example, the following two linked lists:

$$\begin{tikzcd}
A: & & a1 \arrow[r] & a2 \arrow[dr] & & & \\
& & & & c1 \arrow[r] & c2 \arrow[r] & c3 \\
B: & b1 \arrow[r] & b2 \arrow[r] & b3 \arrow[ur] & & &
\end{tikzcd}$$

begin to intersect at node c1.

Notes:

- If the two linked lists have no intersection at all, return null.
- The linked lists must retain their original structure after the function returns.
- You may assume there are no cycles anywhere in the entire linked structure.
- Your code should preferably run in O(n) time and use only O(1) memory.

Credits:

来自 <https://leetcode.com/problems/intersection-of-two-linked-lists/description/>

编写一个程序，找到两个单链表相交的起始节点。

例如，下面的两个链表：

$$\begin{tikzcd}
A: & & a1 \arrow[r] & a2 \arrow[dr] & & & \\
& & & & c1 \arrow[r] & c2 \arrow[r] & c3 \\
B: & b1 \arrow[r] & b2 \arrow[r] & b3 \arrow[ur] & & &
\end{tikzcd}$$

在节点 c1 开始相交。

注意：

- 如果两个链表没有交点，返回 null.
- 在返回结果后，两个链表仍须保持原有的结构。
- 可假定整个链表结构中没有循环。
- 程序尽量满足 O(*n*) 时间复杂度，且仅用 O(*1*) 内存。

Solution for Python3:

```python
# Definition for singly-linked list.
# class ListNode(object):
#     def __init__(self, x):
#         self.val = x
#         self.next = None

class Solution(object):
    def getIntersectionNode(self, headA, headB):
        """
        :type head1, head1: ListNode
        :rtype: ListNode
        """
        p1, p2 = headA, headB
        if not p1 or not p2:
            return None
        while p1 != p2:
            p1 = p1.next
            p2 = p2.next
            if p1 == p2:
                return p1
            if not p1:
                p1 = headB
            if not p2:
                p2 = headA
        return p1
```

Solution for C++:

```
/**
 * 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 *p1 = headA;
        ListNode *p2 = headB;
        if (!p1 || !p2) {
            return NULL;
        }
        while (p1 != p2) {
            p1 = p1->next;
            p2 = p2->next;
            if (p1 == p2) {
                // 有交点：可能是一开始长度相同，中间某个点相遇
                //         可能是到各自链表后在中间相遇
                // 没交点：可能是一开始长度相同，各自均到末尾NULL相等
                //         可能是到各自链表后再到各自末尾NULL相等
                return p1;
            }
            if (!p1) {
                p1 = headB;
            }
            if (!p2) {
                p2 = headA;
            }
        }
        return p1;
    }
};
```