

两个链表的第一个公共结点

作者: [yudake](#)

原文链接: <https://ld246.com/article/1518917658558>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

题目描述

输入两个链表，找出它们的第一个公共结点。

解题思路

两个链表中如果存在公共结点，那么在公共结点之后的所有结点也相同。

- 把两个链表存入两个栈，然后对栈顶进行对比
 - 如果栈顶相同，就将其pop
 - 如果不同，说明上一个结点就是第一个公共节点
 - 如果栈为空，说明此链表的第一个结点就是公共结点

时间复杂度 $O(n+n/2)$ ，空间复杂度 $O(n)$

- 计算两个链表长度，如长度分别为N和M， $N > M$
 - 先对长度为N的链表进行next，直到还剩M个结点
 - 两个链表一起next，对比结点，直到两结点相等

时间复杂度 $O(n)$ ，空间复杂度 $O(1)$

代码

代码一：

```
import java.util.Stack;
public class Solution {
    public ListNode FindFirstCommonNode(ListNode pHead1, ListNode pHead2) {
        if (pHead1 == null || pHead2 == null)
            return null;
        Stack<ListNode> stack1 = new Stack<>();
        Stack<ListNode> stack2 = new Stack<>();
        while (pHead1 != null) {
            stack1.push(pHead1);
            pHead1 = pHead1.next;
        }
        while (pHead2 != null) {
            stack2.push(pHead2);
            pHead2 = pHead2.next;
        }
        ListNode node1 = stack1.pop();
        ListNode node2 = stack2.pop();
        while (node1 == node2 && !stack1.isEmpty() && !stack2.isEmpty()) {
            node1 = stack1.pop();
            node2 = stack2.pop();
        }
        if (node1 == node2)
            return node1;
        return node1.next;
    }
}
```

```
}
```

代码二:

```
public class Solution {
    public ListNode FindFirstCommonNode(ListNode pHead1, ListNode pHead2) {
        if (pHead1 == null || pHead2 == null)
            return null;
        int length1 = getLength(pHead1);
        int length2 = getLength(pHead2);
        if (length1 > length2)
            for (int i = 0; i < length1 - length2; i++)
                pHead1 = pHead1.next;
        if (length2 > length1)
            for (int i = 0; i < length2 - length1; i++)
                pHead2 = pHead2.next;
        while (pHead1 != null && pHead2 != null && pHead1 != pHead2) {
            pHead1 = pHead1.next;
            pHead2 = pHead2.next;
        }
        if (pHead1 == null || pHead2 == null)
            return null;
        return pHead1;
    }

    private int getLength(ListNode node) {
        int i = 0;
        while (node != null) {
            i++;
            node = node.next;
        }
        return i;
    }
}
```