

剑指Offer（五十五）：链表中环的入口结点

🕒 2018年1月24日 11:46:25 📄 4 🌡 5,677 °C 🛠 编辑



一、前言

本系列文章为《剑指Offer》刷题笔记。

刷题平台：[牛客网](#)

书籍下载：[共享资源](#)

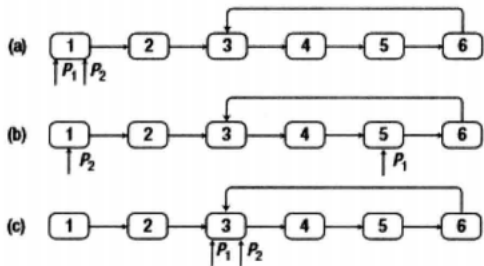
二、题目

一个链表中包含环，请找出该链表的环的入口结点。

1、思路

可以用两个指针来解决这个问题。先定义两个指针P1和P2指向链表的头结点。如果链表中的环有n个结点，指针P1先在链表上向前移动n步，然后两个指针以相同的速度向前移动。当第二个指针指向的入口结点时，第一个指针已经围绕着凑了一圈又回到了入口结点。

以下图为例，指针P1和P2在初始化时都指向链表的头结点。由于环中有4个结点，指针P1先在链表上向前移动4步。接下来两个指针以相同的速度在链表上向前移动，直到它们相遇。它们相遇的结点正好是环的入口结点。



现在，关键在于怎么知道环中有几个结点呢？

可以使用快慢指针，一个每次走一步，一个每次走两步。如果两个指针相遇，表明链表中存在环，并且两个指针相遇的结点一定在环中。

随后，我们就从相遇的这个环中结点出发，一边继续向前移动一边计数，当再次回到这个结点时，就可以得到环中结点数目了。

2、代码

C++:

```
1  /*
2  struct ListNode {
3      int val;
4      struct ListNode *next;
5      ListNode(int x) :
6          val(x), next(NULL) {
7      }
8  };
9  */
10 class Solution {
11 public:
```


```

12  ListNode* EntryNodeOfLoop(ListNode* pHead)
13  {
14      if(pHead == NULL){
15          return NULL;
16      }
17      ListNode* meetingnode = MeetingNode(pHead);
18      if(meetingnode == NULL){
19          return NULL;
20      }
21      // 回环链表结点数
22      int nodesloop = 1;
23      // 找到环中结点数
24      ListNode* pNode1 = meetingnode;
25      while(pNode1->next != meetingnode){
26          pNode1 = pNode1->next;
27          nodesloop++;
28      }
29      pNode1 = pHead;
30      // 第一个指针向前移动nodesloop步
31      for(int i = 0; i < nodesloop; i++){
32          pNode1 = pNode1->next;
33      }
34      // 两个指针同时移动，找到环入口
35      ListNode* pNode2 = pHead;
36      while(pNode1 != pNode2){
37          pNode1 = pNode1->next;
38          pNode2 = pNode2->next;
39      }
40      return pNode1;
41  }
42  private:
43      // 使用快慢指针，找到任意的一个环中结点
44      ListNode* MeetingNode(ListNode* pHead){
45          ListNode* pSlow = pHead->next;
46          if(pSlow == NULL){
47              return NULL;
48          }
49          ListNode* pFast = pSlow->next;
50          while(pFast != NULL && pSlow != NULL){
51              if(pFast == pSlow){
52                  return pFast;
53              }
54              pSlow = pSlow->next;
55              pFast = pFast->next;
56              if(pFast != NULL){
57                  pFast = pFast->next;
58              }
59          }
60          return NULL;
61      }
62  };

```

Python:

	Pytho
<pre> 1 # -*- coding:utf-8 -*- 2 # class ListNode: 3 # def __init__(self, x): 4 # self.val = x 5 # self.next = None 6 class Solution: 7 def EntryNodeOfLoop(self, pHead): 8 # write code here 9 if pHead == None: 10 return None 11 meetingnode = self.MeetingNode(pHead) 12 if meetingnode == None: 13 return None 14 nodeslop = 1 15 node1 = meetingnode 16 while node1.next != meetingnode: 17 node1 = node1.next 18 nodeslop += 1 19 node1 = pHead 20 for _ in range(nodeslop): 21 node1 = node1.next 22 node2 = pHead 23 while node1 != node2: 24 node1 = node1.next 25 node2 = node2.next 26 return node1 27 28 def MeetingNode(self, pHead): 29 slow = pHead.next 30 if slow == None: 31 return None 32 fast = slow.next 33 while fast != None and slow != None: 34 if slow == fast: 35 return fast 36 slow = slow.next 37 fast = fast.next 38 if fast != None: 39 fast = fast.next 40 return None </pre>	



微信公众号

分享技术，乐享生活：微信公众号搜索「JackCui-AI」关注一个在互联网摸爬滚打的潜行者。

君子欲讷于言而敏于行。--- 孔子