removeListNode

2016 年 4 月 5 日

1 删除链表中的元素

删除链表中等于给定值 val 的所有节点。

样例 给出链表 1->2->3->3->4->5->3, 和 val = 3, 你需要返回删除 3 之后的链表：1->2->4->5。

```
In [10]: # Definition for singly-linked list.
         class ListNode:
             def __init__(self, x, next=None):
                 self.val = x
                 self.next = next

             def travel(self):
                 while self != None:
                     print self.val, "->",
                     self = self.next
                 print "null"

         class Solution:
             # @param head, a ListNode
             # @param val, an integer
             # @return a ListNode
             def removeElements(self, head, val):
                 # Write your code here

                 # Firstly, result direct the head of the linklist
                 result = head
                 if head == None:
                     return None

                 # Secondly, delete the val in the linklist except the first element
```

```
        while head.next != None:
            if head.next.val == val:
                if head.next.next != None:
                    head.next = head.next.next
                else:
                    head.next = None
                    break
            else:
                head = head.next

        # Thirdly, delete the first element if it equals to val
        if result.val == val:
            result = result.next
        return result

L1 = ListNode(1, ListNode(2, ListNode(3, ListNode(3, ListNode(4, ListNode(5, ListNode(3,
#L1 = ListNode(1)
L1.travel()
L2 = L1
#
sol = Solution()
L1 = sol.removeElements(L1, 3)
#
L1.travel()
L2.travel()
#
L3 = ListNode(1, ListNode(1, ListNode(2)))
L3 = sol.removeElements(L3, 1)
L3.travel()
```

```
1 -> 2 -> 3 -> 3 -> 4 -> 5 -> 3 -> null
1 -> 2 -> 4 -> 5 -> null
1 -> 2 -> 4 -> 5 -> null
2 -> null
```