206 Reverse Linked List

2018年4月4日 21:04

Question:

Reverse a singly linked list.

click to show more hints.

Hint:

A linked list can be reversed either iteratively or recursively. Could you implement both?

来自 <https://leetcode.com/problems/reverse-linked-list/description/>

反转一个单链表。

进阶:

链表可以迭代或递归地反转。你能否两个都实现一遍？

Solution for Python3:

```
1   # Definition for singly-linked list.
2   # class ListNode:
3   #     def __init__(self, x):
4   #         self.val = x
5   #         self.next = None
6   # Iterative Version
7   class Solution1:
8       def reverseList(self, head):
9           """
10          :type head: ListNode
11          :rtype: ListNode
12          """
```

```
13          if not head:
14              return head
15          cur = head
16          pos = head.next
17          cur.next = None
18          while pos:
19              head = pos
20              pos = head.next
21              head.next = cur
22              cur = head
23          return head
24
25  # Recursive Version
26  class Solution:
27      def reverseList(self, head):
28          """
29          :type head: ListNode
30          :rtype: ListNode
31          """
32          if not head or not head.next:
33              return head
34          P = self.reverseList(head.next)
35          head.next.next = head
36          head.next = None
37          return P
```

Solution for C++:

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
// Iterative Version
class Solution {
public:
    ListNode* reverseList(ListNode* head) {
        if (!head) {
            return head;
        }
        ListNode *cur = head;
        ListNode *pos = head->next;
        for (cur->next = NULL; pos != NULL;) {
            head = pos;
            pos = head->next;
            head->next = cur;
            cur = head;
        }
        return head;
    }
};

// Recursive Version
class Solution {
public:
    ListNode* reverseList(ListNode* head) {
        if (!head or !head->next) {
            return head;
```

```
34            }
35            ListNode* P = reverseList(head->next);
36            head->next->next = head;
37            head->next = NULL;
38            return P;
39        }
40    };
```