206. Reverse Linked List

题目描述：

> 将一个单链表反转

解题思路：

利用迭代或者循环的方法处理每个节点，依次转换。

注意：一定要考虑空链表和只有一个节点的链表的情况！！

代码（Iteratively）：

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    ListNode* reverseList(ListNode* head) {
        if(head == NULL||head->next==NULL)
            return head;
        ListNode *now, *ne1, *ne2;
        now = head;
        ne1 = now -> next;
        while(ne1 != NULL){
            ne2 = ne1->next;
            ne1->next = now;
            now = ne1;
            ne1 = ne2;
        }
        head->next = NULL;
        head = now;
        return head;
    }
};
```

代码（Recursively）：

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    ListNode* reverseList(ListNode* head) {
        if(head == NULL)
            return head;
        return List(head, NULL, head->next);
    }
    ListNode* List(ListNode *now, ListNode *last, ListNode *ne){
        if(ne == NULL){
            now->next = last;
            return now;
        }
        now->next = last;
        return List(ne, now, ne->next);
    }
};
```