一、题目说明

题目206. Reverse Linked List，翻转一个链表。难度是Easy！可以递归，也可以非递归。

二、我的解答

翻转，这个确实不难，非递归算法：

```
class Solution{
    public:
        ListNode* reverseList(ListNode* head){
            if(head == NULL || head->next==NULL) return head;
            ListNode *cur,*p = head->next;
            bool first = true;
            while(p !=NULL && p->next!=NULL){
                cur = p->next;

                p->next = head;
                if(first){
                    head->next = NULL;
                    first = false;
                }

                head = p;
                p = cur;
            }
            if(p !=NULL){
                p->next = head;
                if(first){
                    head->next = NULL;
                    first = false;
                }
                head = p;
            }

            return head;
        }
};
```

性能如下：

```
Runtime: 8 ms, faster than 88.11% of C++ online submissions for Reverse Linked
List.
Memory Usage: 9.7 MB, less than 5.34% of C++ online submissions for Reverse
Linked List.
```

三、优化措施

用翻转链表法实现：

```
class Solution{
    public:
        ListNode* reverseList(ListNode* head){
            //翻转链表
            ListNode *pre = NULL;
            ListNode *cur = head;
            ListNode *tmp = NULL;
            while(cur!=NULL) {
                //记录当前节点的下一个节点
                tmp = cur->next;
                //然后将当前节点指向pre
                cur->next = pre;
                //pre和cur节点都前进一位
                pre = cur;
                cur = tmp;
            }
            return pre;
        }
};
```

```
Runtime: 12 ms, faster than 38.84% of C++ online submissions for Reverse Linked
List.
Memory Usage: 9.7 MB, less than 5.34% of C++ online submissions for Reverse
Linked List.
```