1. Swap Nodes in Pairs

Medium

Given a linked list, swap every two adjacent nodes and return its head.

You may **not** modify the values in the list’s nodes, only nodes itself may be changed.

**Example:**

Given 1->2->3->4, you should return the list as 2->1->4->3.

**分析：**递归操作。源于翻转应该从后往前进行

/\*\*  
 \* Definition for singly-linked list.  
 \* struct ListNode {  
 \* int val;  
 \* ListNode \*next;  
 \* ListNode(int x) : val(x), next(NULL) {}  
 \* };  
 \*/  
class Solution {  
public:  
 ListNode\* swapPairs(ListNode\* head) {  
 if(head == NULL || head->next == NULL)return head;  
 ListNode \*p = swapPairs(head->next->next);  
 ListNode \*temp = head->next;  
 head->next = p;  
 temp->next = head;  
 return temp;  
 }  
};