

## k个元素一组翻转链表

给定一个链表，对链表的每k个元素一组进行翻转，如果剩余长度不满足k，则顺序不变。

分析： 每次找到k个元素，记开始为a,结束为b， 每次翻转[a,b)之间的元素，长度不够的时候直接返回。

```
class Solution {
public:
    ListNode* reverseKGroup(ListNode* head, int k) {
        if (head == nullptr || k <= 0) {
            return head;
        }

        ListNode *a, *b; // a逆置之后为每个k段的最后一个节点，b为下一个段的开始节点
        a = b = head;
        for (int i = 0; i < k; i++) {
            if (b == nullptr) {
                return head;
            }
            b = b->next;
        }

        ListNode* newHead = reverse(a, b);
        a->next = reverseKGroup(b, k);

        return newHead;
    }

private:
    // 翻转[a,b)的元素
    ListNode* reverse(ListNode* a, ListNode* b) {
        ListNode *pre = nullptr, *curr = a, *next;
        while (curr != b) {
            next = curr->next;
            curr->next = pre;
            pre = curr;
            curr = next;
        }

        return pre;
    }
};
```