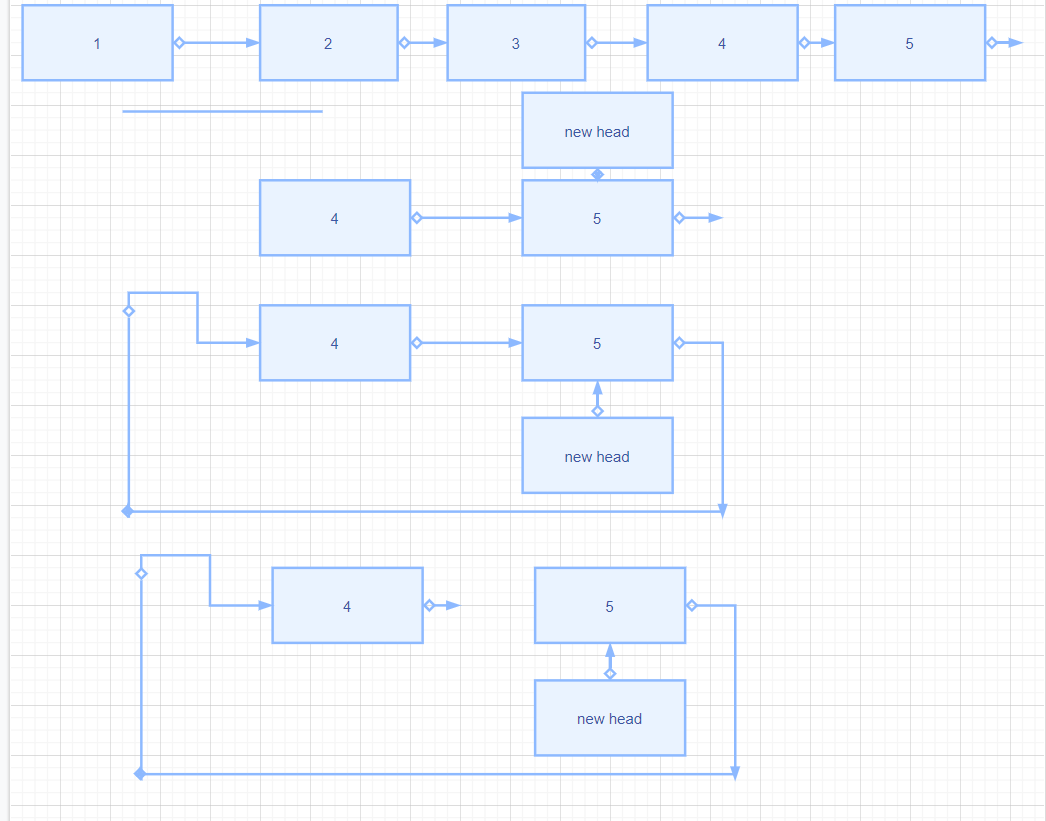
题目描述：

翻转一个链表

解题思路：

要翻转一个链表，第一个想法就是递归。要找到链表的最后一个节点和倒数第二个节点，把这两个顺序翻转过来以后又要去找倒数第三个节点，因此我第一个想法就是用递归的手法来解这道题。

我个人画的图解大概是这样子的：



ListNode\* reverse(ListNode\* head) {

// write your code here

if (head->next == nullptr || head == NULL) {

return head;

}

else {

ListNode\* new\_head = reverse(head->next);

head->next->next = head;

head->next = nullptr;

return new\_head;

}

}

代码中第一部分是递归终止条件。

而else中的第二步最终会形成一个环形链表，需要重置head->next的指向。这一步也是最容易犯的错误。

截图如下：

