1. Reorder List

Given a singly linked list *L*: *L*0→*L*1→…→*L****n-1→Ln, reorder it to: L0→L****n*→*L*1→*L****n-1→L2→L****n*-2→…

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

**Example 1:**

Given 1->2->3->4, reorder it to 1->4->2->3.

**Example 2:**

Given 1->2->3->4->5, reorder it to 1->5->2->4->3.

**解**

分为3个步骤

* 寻找中点：双指针
* 翻转后一半链表：三指针法
* 合并：归并

/\*\*  
 \* Definition for singly-linked list.  
 \* struct ListNode {  
 \* int val;  
 \* ListNode \*next;  
 \* ListNode() : val(0), next(nullptr) {}  
 \* ListNode(int x) : val(x), next(nullptr) {}  
 \* ListNode(int x, ListNode \*next) : val(x), next(next) {}  
 \* };  
 \*/  
class Solution {  
public:  
 void reorderList(ListNode\* head) {  
 if(head == NULL || head->next == NULL)return;  
 ListNode \*p = head, \*pp = head;  
 while(pp && pp->next){  
 p = p->next;  
 pp = pp->next->next;  
 }  
 ListNode \*pre = NULL, \*cur = p, \*next;  
 while(cur){  
 next = cur->next;  
 cur->next = pre;  
 pre = cur;  
 cur = next;  
 }  
 pp = head, p = pre;  
 ListNode \*tmp1, \*tmp2;  
 while(p->next){  
 tmp1 = pp->next;  
 tmp2 = p->next;  
 pp->next = p;  
 p->next = tmp1;  
   
 pp = tmp1;  
 p = tmp2;  
 }  
 }  
};