https://leetcode.com/problems/reorder-list/

```
/**
 * 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:
    ListNode* reverseList(ListNode* head){
        ListNode* p = nullptr;
        ListNode* c = head;
        ListNode* n = head;

        while (n){
            n = n->next;
            c->next = p;
            p = c;
            c = n;
        }

        return p;
    }
    void reorderList(ListNode* head) {
        ListNode* slow = head;
        ListNode* fast = head;

        while (fast && fast->next){
            slow = slow->next;
            fast = fast->next->next;
        }

        ListNode* l2 = slow->next;
        l2 = reverseList(l2);
        slow->next = nullptr;
        ListNode* l1 = head;


        ListNode* newList = nullptr, *itr = nullptr;
        newList = l1;
        l1 = l1->next;
        itr = newList;

        while (l1 != nullptr && l2 != nullptr){
            itr->next = l2;
            l2 = l2->next;
            itr = itr->next;

            itr->next = l1;
            l1 = l1->next;
            itr = itr->next;
        }

        while (l1){
            itr->next = l1;
            l1 = l1->next;
            itr = itr->next;
        }
        while (l2){
            itr->next = l2;
            l2 = l2->next;
            itr = itr->next;
        }
    }
};
```