234. Palindrome Linked List

题目描述：https://leetcode.com/problems/palindrome-linked-list/

> 判断一个链表是不是回文的

解题思路：

找到链表的中点，将后半段逆序，然后以此判断是否相同

代码：

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    ListNode* reverseList(ListNode *head){
        ListNode *fakeHead = NULL;
        ListNode *pre = fakeHead;
        ListNode *now = head;
        while(now != NULL){
            ListNode * tmp = now->next;
            now->next = pre;
            pre = now;
            now = tmp;
        }
        return pre;
    }
    bool isPalindrome(ListNode* head) {
        if(head==NULL)
            return true;
        ListNode *fast = head->next;
        ListNode *slow = head;
        while(fast != NULL && fast->next != NULL){
            fast = fast->next;
            fast = fast->next;
            slow = slow->next;
        }
```

```
        if(fast != NULL){
            slow = slow->next;
        }
        cout<<"slow:"<<slow->val<<endl;
        ListNode * rhead = reverseList(slow);
        cout<<"rhead:"<<rhead->val<<endl;
        while(rhead && head){
            if(rhead->val == head->val){
                rhead = rhead->next;
                head = head->next;
            }
            else{
                return false;
            }
        }
        return true;
    }
};
```