[每日 LeetCode] 234. Palindrome Linked List

作者：Hanseltu
原文链接：https://ld246.com/article/1556632563204
来源网站：链滴

Description:

Given a singly linked list, determine if it is a palindrome.

Example 1:

Input: 1->2
Output: false

Example 2:

Input: 1->2->2->1
Output: true

Follow up:

Could you do it in O(n) time and O(1) space?

思路：本题要求判断链表是否为回文链表。最开始想到的是使用栈，将链表的前半部分依次进栈，然依次出栈和后半部分的值比较。由于空间复杂度要求O(1)，使用原地反转链表思想，将后半部分的链原地反转，然后依次和前面的元素比较。使用了[每日LeetCode] 206. Reverse Linked List的反转链代码。

C++代码

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    bool isPalindrome(ListNode* head) {
        if(!head||!head->next) return true;

        ListNode* fast;
        ListNode* slow;
        fast = slow = head;
        while(fast&&fast->next){
            slow = slow->next;
            fast = fast->next->next;
        }
        if(fast){
            slow = slow->next;
        }

        slow = reverseList(slow);
```

```
        while(slow){
            if(head->val != slow->val) return false;
            head = head->next;
            slow = slow->next;
        }
        return true;
    }

    ListNode* reverseList(ListNode* head) {
        if(head == NULL || head->next == NULL)
            return head;

        ListNode* base = reverseList(head->next);
        head->next->next = head;
        head->next = NULL;
        return base;
    }
};
```

运行时间：36ms

运行内存：13.6M