234. Palindrome Linked List

link

判断回文这里用的是取终点然后翻转后面, 判断跟前面是不是相同.

注意快慢指针偶数拿第一个得fast.next && fast.next.next != null; 143题那块是fast和fast.next实际有点小问题, 但是能过是因为题目本来就要把那个加到最后.

```
    public boolean isPalindrome(ListNode head) {
        if(head == null || head.next == null) return true;
        ListNode slow = head, fast = head, newHead = null, next = null;
        while(fast.next != null && fast.next.next != null){
            fast = fast.next.next;
            slow = slow.next;
        }
        fast = slow.next;
        slow.next = null;
        while(fast != null){
            next = fast.next;
            fast.next = newHead;
            newHead = fast;
            fast = next;
/*
        ListNode before = reverse;
        reverse = slow;
        slow = slow.next;
        revvers.next = before;
*/
        }
        fast = newHead;
        while(fast != null){
            if(fast.val != head.val)
                return false;
            fast = fast.next;
            head = head.next;
        }
        return true;
    }
```

实际上有更好的解决方案, 就是fast开始判断的时候 可以直接翻转slow, 最好之后再翻转回去.

```
def isPalindrome(self, head):
    rev = None
    fast = head
    while fast and fast.next:
        fast = fast.next.next
        rev, rev.next, head = head, rev, head.next
    tail = head.next if fast else head
```

```
    isPali = True
    while rev:
        isPali = isPali and rev.val == tail.val
        head, head.next, rev = rev, head, rev.next
        tail = tail.next
    return isPali
```

```
ListNode* tmp = rev;
rev = slow;
slow = slow -> next;
rev -> next = tmp;
```