public class Solution {

public boolean isPalindrome(ListNode head) {

if(head==null||head.next==null) return true;

ListNode sl=head;

ListNode qk=head;

while(qk!=null&&qk.next!=null){

sl=sl.next;

qk=qk.next.next;

}

//反转后半段,再顺次比较；

ListNode root=new ListNode(0);

ListNode n=null;

while(sl!=null){

n=sl.next;

sl.next=root.next;

root.next=sl;

sl=n;

}

root=root.next;

while(root!=null&&head!=null){

if(head.val!=root.val){

return false;

}

head=head.next;

root=root.next;

}

return true;

/\*ListNode root=head;//不可链表反转后比较，因为即使保留头指针，反转后链表就已经改变了无法与原来链表对照；

ArrayList<Integer> l=new ArrayList<Integer>();//用list暂存超时，尝试用栈存前半段，与后半段比较，但题目要求用O(1)空间；

while(root!=null){

l.add(root.val);

root=root.next;

}

ListNode nroot=new ListNode(0);

ListNode n=null;

while(head!=null){

n=head.next;

head.next=nroot.next;

nroot.next=head;

head=n;

}

nroot=nroot.next;

int i=0;

while(nroot!=null){

if(l.get(i)!=nroot.val){

return false;

}

i++;

nroot=nroot.next;

}

return true;\*/

}