

回文链表

请判断一个链表是否为回文链表。

示例 1:

输入: 1->2

输出: false

示例 2:

输入: 1->2->2->1

输出: true

```
/**
 * 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:
    //率先进行快慢指针找到中间分隔 分为前半部和后半部
    ListNode* findlist(ListNode* head)
    {
        ListNode* fast=head;
        ListNode* slow=head;
        while(fast->next!=nullptr&&fast->next->next!=nullptr)
        {
            fast=fast->next->next;
            slow=slow->next;
        }
        return slow;
    }
    //链表反转
    ListNode* reverse(ListNode* head)
    {
        ListNode* prev=nullptr;
        ListNode* cur=head;
        while(cur!=nullptr)
        {
```

```
        ListNode* tmp=cur->next;//保留后续不丢失
        cur->next=prev;//改变方向
        prev=cur;
        cur=tmp;
    }
    return prev;
}
bool isPalindrome(ListNode* head) {
    if(head==nullptr)
        return true;
    ListNode* cur=head;
    ListNode* part2=findlist(head);
    ListNode* partlist=reverse(part2->next);

    while(partlist!=nullptr)
    {
        if(cur->val!=partlist->val)
            return false;
        cur=cur->next;
        partlist=partlist->next;
    }
    return true;
}
};
```