

链表回文判断

对于一个链表，请设计一个时间复杂度为 $O(n)$, 额外空间复杂度为 $O(1)$ 的算法，判断其是否为回文结构。

给定一个链表的头指针 A，请返回一个 bool 值，代表其是否为回文结构。保证链表长度小于等于 900。

测试样例：

1->2->2->1

返回：true

```
/*
struct ListNode {
    int val;
    struct ListNode *next;
    ListNode(int x) : val(x), next(NULL) {}
};*/
class PalindromeList {
public:
    //链表反转
    struct ListNode* reverseList(struct ListNode* head)
    {
        if(head==NULL)
            return NULL;

        struct ListNode* prev=NULL;
        struct ListNode* cur=head;
        while(cur)
        {
            struct ListNode* tmp=cur->next;
            cur->next=prev;
            prev=cur;
            cur=tmp;
        }
        return prev;
    }
    bool chkPalindrome(ListNode* A) {
        // write code here
        //快慢指针找到中间节点
        ListNode* slow=A;
        ListNode* fast=A;
        while(fast!=NULL&&fast->next!=NULL)
        {
            slow=slow->next;
            fast=fast->next->next;
        }
    }
};
```

```
    }  
    ListNode* p=reverseList(slow);//从中间节点到结尾后半段反转  
    while(p!=NULL&&A!=NULL)  
    {  
        if(p->val!=A->val)  
        {  
            return false;  
        }  
        else  
        {  
            p=p->next;  
            A=A->next;  
        }  
    }  
    return true;  
}  
};
```