234. Palindrome Linked List

Add to List

Question | Editorial Solution | My Submissions (/problems/palindrome-linked-list/submissions/)

Total Accepted: **83924** Total Submissions: **264759** Difficulty: **Easy** Contributors: **Admin**

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

Follow up:

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

Subscribe (/subscribe/) to see which companies asked this question

Hide Tags | Linked List (/tag/linked-list/) | Two Pointers (/tag/two-pointers/)

Hide Similar Problems | (E) Palindrome Number (/problems/palindrome-number/) | (E) Valid Palindrome (/problems/valid-palindrome/) | (E) Reverse Linked List (/problems/reverse-linked-list/)

Have you met this question in a real interview? Yes No

Discuss (https://discuss.leetcode.com/category/290) | Top Solutions | Pick One (/problems/random-one-question/)

C++

```
/**
 * 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) {
        ListNode* slow = head;
        ListNode* fast = head;
        while(fast!=NULL&&fast->next!=NULL){
            fast = fast->next->next;
            slow = slow->next;
        }
        if(fast!=NULL) slow = slow->next;
        ListNode* reverse = NULL;
        while(slow!=NULL){
            fast = slow;
            slow = slow->next;
            fast->next = reverse;
            reverse = fast;
        }
        while(reverse!=NULL){
            if(reverse->val!=head->val) return false;
            reverse = reverse->next;
            head = head->next;
        }
        return true;
    }
};
```

Custom Testcase

Contribute Testcase

Run Code | Submit Solution

Send Feedback (mailto:admin@leetcode.com?subject=Feedback)