Problem 234: Palindrome Linked List

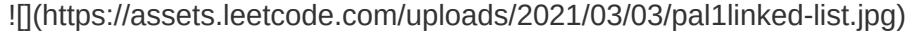
Problem Information

Difficulty: Easy
Acceptance Rate: 56.90%
Paid Only: No
Tags: Linked List, Two Pointers, Stack, Recursion

Problem Description

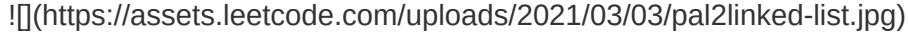
Given the `head` of a singly linked list, return `true` _if it is a_ _palindrome_ _or_ `false` _otherwise_.

Example 1:



Input: head = [1,2,2,1] **Output:** true

Example 2:



Input: head = [1,2] **Output:** false

Constraints:

* The number of nodes in the list is in the range `[1, 105]`. * `0 <= Node.val <= 9`

Follow up: Could you do it in `O(n)` time and `O(1)` space?

Code Snippets

C++:

```
/**
* 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:
bool isPalindrome(ListNode* head) {

}
};
```

Java:

```
/**
* Definition for singly-linked list.
* public class ListNode {
* int val;
* ListNode next;
* ListNode() {}
* ListNode(int val) { this.val = val; }
* ListNode(int val, ListNode next) { this.val = val; this.next = next; }
* }
*/
class Solution {
public boolean isPalindrome(ListNode head) {

}
}
```

Python3:

```
# Definition for singly-linked list.
# class ListNode:
# def __init__(self, val=0, next=None):
# self.val = val
# self.next = next
class Solution:
```

```
def isPalindrome(self, head: Optional[ListNode]) -> bool:
```