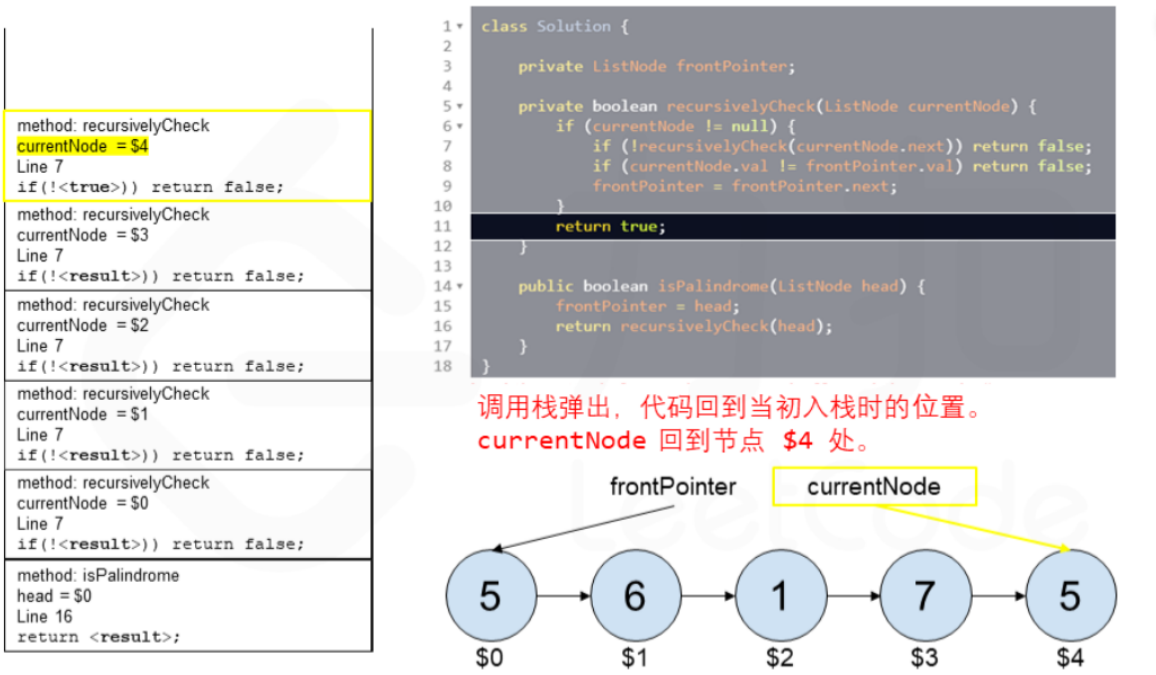
234. Palindrome Linked List

My Solution

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

        stack = []
        while head != None:
            stack.append(head.val)
            head = head.next

        return stack == stack[::-1]
```

I tried to reverse the linked list, but it did not work. Because once I reversed it, the head node will change too.

Recursion Solution



method: recursivelyCheck
currentNode = $3
Line 7
`if(!<result>)) return false;`

method: recursivelyCheck
currentNode = $2
Line 7
`if(!<result>)) return false;`

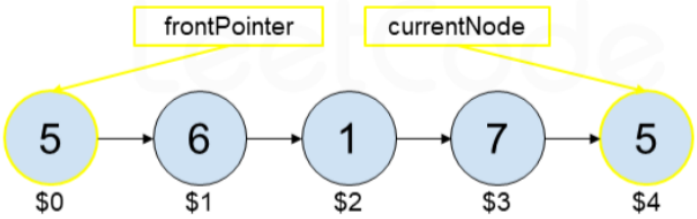
method: recursivelyCheck
currentNode = $1
Line 7
`if(!<result>)) return false;`

method: recursivelyCheck
currentNode = $0
Line 7
`if(!<result>)) return false;`

method: isPalindrome
head = $0
Line 16
`return <result>;`

```java
class Solution {

    private ListNode frontPointer;

    private boolean recursivelyCheck(ListNode currentNode) {
        if (currentNode != null) {
            if (!recursivelyCheck(currentNode.next)) return false;
            if (currentNode.val != frontPointer.val) return false;
            frontPointer = frontPointer.next;
        }
        return true;
    }

    public boolean isPalindrome(ListNode head) {
        frontPointer = head;
        return recursivelyCheck(head);
    }
}
```

由于 `frontPointer` 和 `currentNode` 的值相等，因此 `if` 函数体不执行，继续来看下一行代码。



method: recursivelyCheck
currentNode = $3
Line 7
`if(!<true>)) return false;`

method: recursivelyCheck
currentNode = $2
Line 7
`if(!<result>)) return false;`

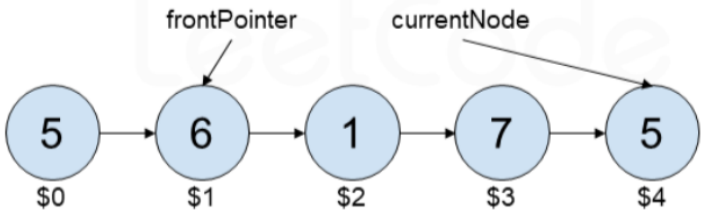
method: recursivelyCheck
currentNode = $1
Line 7
`if(!<result>)) return false;`

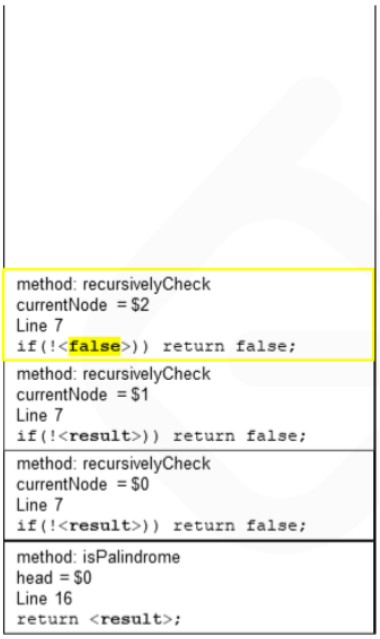
method: recursivelyCheck
currentNode = $0
Line 7
`if(!<result>)) return false;`

method: isPalindrome
head = $0
Line 16
`return <result>;`

```java
class Solution {

    private ListNode frontPointer;

    private boolean recursivelyCheck(ListNode currentNode) {
        if (currentNode != null) {
            if (!recursivelyCheck(currentNode.next)) return false;
            if (currentNode.val != frontPointer.val) return false;
            frontPointer = frontPointer.next;
        }
        return true;
    }

    public boolean isPalindrome(ListNode head) {
        frontPointer = head;
        return recursivelyCheck(head);
    }
}
```

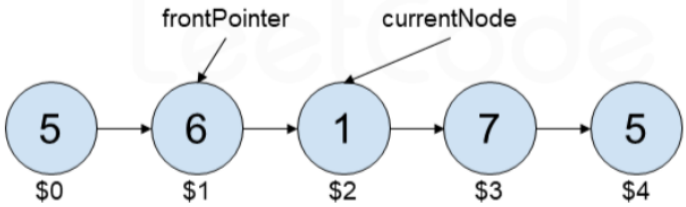
然后返回 `true`，类似地，它的值填充到栈顶的信息中。






```
class Solution {

    private ListNode frontPointer;

    private boolean recursivelyCheck(ListNode currentNode) {
        if (currentNode != null) {
            if (!false)) return false;
            if (currentNode.val != frontPointer.val) return false;
            frontPointer = frontPointer.next;
        }
        return true;
    }

    public boolean isPalindrome(ListNode head) {
        frontPointer = head;
        return recursivelyCheck(head);
    }
}
```

然后我们回到 `if` 语句中。



result keeps staying for False.

```python
class Solution:
    def isPalindrome(self, head: ListNode) -> bool:

        self.front_pointer = head

        def recursively_check(current_node=head):
            if current_node is not None:
                if not recursively_check(current_node.next):
                    return False
                if self.front_pointer.val != current_node.val:
                    return False
                self.front_pointer = self.front_pointer.next
            return True

        return recursively_check()
```

author: LeetCode-Solution
Link: https://leetcode.cn/problems/palindrome-linked-list/solution/hui-wen-lian-biao-by-leetcode-solution/