

[Description](#)
[Solution](#)
[Discuss \(999+\)](#)
[Submissions](#)

isPalindrome may, you could return false right when one of the comparisons fail.

```
var reverseList = function (head) {
    var prev = null;
    while(head) {
        var next = head.next;
        head.next = prev;
        prev = head;
        head = next;
    }
    return prev;
}

var findMiddleNode = function (head) {
    if(head === null) return null;

    var slow = head;
    var fast = head;
    while(fast.next) {
        fast = fast.next;
        slow = slow.next;
        if(fast.next) return slow;
        fast = fast.next;
    }
    return slow;
}

var isPalindrome = function(head) {
    var middle = findMiddleNode(head);

    var reverseHalf = reverseList(middle);

    var current = head;
    var currentReverse = reverseHalf;

    var isPalindrome = true;

    while(current.next) {
        // change to return false; if you don't care about the original list.
        if(current.val !== currentReverse.val) isPalindrome = false;
        current = current.next;
        currentReverse = currentReverse.next;
    }
    current.next = reverseList(currentReverse);

    return isPalindrome;
};
```