

Problem Challenge 1

We'll cover the following

- Palindrome LinkedList (medium)
- Try it yourself

Palindrome LinkedList (medium)

Given the head of a **Singly LinkedList**, write a method to check if the **LinkedList is a palindrome** or not.

Your algorithm should use **constant space** and the input LinkedList should be in the original form once the algorithm is finished. The algorithm should have $O(N)$ time complexity where 'N' is the number of nodes in the LinkedList.

Example 1:


```
Input: 2 -> 4 -> 6 -> 4 -> 2 -> null
Output: true
```


Example 2:


```
Input: 2 -> 4 -> 6 -> 4 -> 2 -> 2 -> null
Output: false
```


Try it yourself

Try solving this question here:

 Java

 Python3

 JS


 C++

```
1
2 class ListNode {
3     int value = 0;
4     ListNode next;
5
6     ListNode(int value) {
7         this.value = value;
8     }
9 }
10
11 class PalindromicLinkedList {
12
13     public static boolean isPalindrome(ListNode head) {
14         // TODO: Write your code here
15         return false;
16     }
17
18     public static void main(String[] args) {
19         ListNode head = new ListNode(2);
20         head.next = new ListNode(4);
21         head.next.next = new ListNode(6);
22         head.next.next.next = new ListNode(4);
23         head.next.next.next.next = new ListNode(2);
24         System.out.println("Is palindrome: " + PalindromicLinkedList.isPalindrome(head));
25
26         head.next.next.next.next.next = new ListNode(2);
27         System.out.println("Is palindrome: " + PalindromicLinkedList.isPalindrome(head));
28     }
29 }
```

Run

Save

Reset





← Back

Next →

Middle of the LinkedList (easy)

Solution Review: Problem Challenge 1

✓ Completed

 Report an Issue  Ask a Question

