**[Palindrome Linked List](https://leetcode.com/problems/palindrome-linked-list/)**

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?

class Solution {

public:

    bool isPalindrome(ListNode\* head) {

        vector<int> listVals;

        while (head) {

            listVals.push\_back(head->val);

            head = head->next;

        }

        int left = 0, right = listVals.size() - 1;

        while (left < right && listVals[left] == listVals[right]) {

            left++;

            right--;

        }

        return left >= right;

    }

};

Link : <https://leetcode.com/problems/palindrome-linked-list/description/?envType=daily-question&envId=2024-03-22>