Problem Link:

<https://leetcode.com/problems/palindrome-linked-list/>

Solution:

/\*\*

\* 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) {

if(!head || !head->next)

return true;

ListNode\* slow = head;

ListNode\* fast = head;

while(fast && fast->next)

{

slow = slow->next;

fast = fast->next->next;

}

ListNode\* prev = nullptr;

ListNode\* curr = slow;

while(curr)

{

ListNode\* nextTemp = curr->next;

curr->next = prev;

prev = curr;

curr = nextTemp;

}

ListNode\* first = head;

ListNode\* second = prev;

while(second)

{

if(first->val != second->val)

return false;

first = first->next;

second = second->next;

}

return true;

}

};