

Problem List

Submit

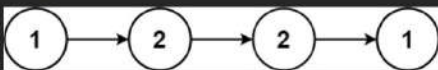
Premium

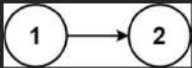
DescriptionEditorialSolutionsSubmissions

234. Palindrome Linked List

EasyTopicsCompanies

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:

18K357167 Online

Code

```
6  *};
7  */
8  struct ListNode* reverselist(struct ListNode* head) {
9      struct ListNode* prev = NULL;
10     struct ListNode* curr = head;
11     struct ListNode* nextNode;
12
13     while (curr != NULL) {
14         nextNode = curr->next;
15         curr->next = prev;
16         prev = curr;
17         curr = nextNode;
18     }
19     return prev;
20 }
21
22 bool isPalindrome(struct ListNode* head) {
23     if (head == NULL || head->next == NULL)
24         return true;
25
26     struct ListNode *slow = head, *fast = head;
27
28     while (fast != NULL && fast->next != NULL) {
29         slow = slow->next;
30     }
31
32     struct ListNode* prev = NULL;
33     struct ListNode* curr = slow;
34     while (curr != NULL) {
35         nextNode = curr->next;
36         curr->next = prev;
37         prev = curr;
38         curr = nextNode;
39     }
40
41     struct ListNode* p1 = head;
42     struct ListNode* p2 = prev;
43     while (p1 != NULL && p2 != NULL) {
44         if (p1->val != p2->val)
45             return false;
46         p1 = p1->next;
47         p2 = p2->next;
48     }
49     return true;
50 }
```

Accepted Runtime: 0 ms

Activate Windows
Go to Settings to activate Windows.

Case 1 Case 2

Problem List

Submit

0

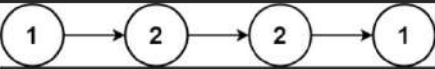
Premium

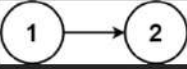
DescriptionEditorialSolutionsSubmissions

234. Palindrome Linked List

EasyTopicsCompanies

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:

18K357166 Online

Code

```
21
22 bool isPalindrome(struct ListNode* head) {
23     if (head == NULL || head->next == NULL)
24         return true;
25
26     struct ListNode *slow = head, *fast = head;
27
28     while (fast != NULL && fast->next != NULL) {
29         slow = slow->next;
30         fast = fast->next->next;
31     }
32
33     struct ListNode* secondHalf = reverseList(slow);
34
35     struct ListNode* p1 = head;
36     struct ListNode* p2 = secondHalf;
37
38     while (p2 != NULL) {
39         if (p1->val != p2->val)
40             return false;
41         p1 = p1->next;
42         p2 = p2->next;
43     }
44 }
```

Accepted Runtime: 0 ms

Case 1Case 2

Activate Windows
Go to Settings to activate Windows.