3i-DS-SEP-2025 | Remove Nth Node From End of | Linked List Cycle - LeetCode | Palindrome Linked List - LeetC | New announcement "todays l

leetcode.com/problems/linked-list-cycle/description/?envType=study-plan-v2&envId=top-interview-150

Top Interview 150 | Submit | Premium

Description | Accepted | Editorial | Solutions | Submissions

141. Linked List Cycle

Solved

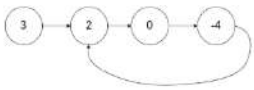
Easy | Topics | Companies

Given `head`, the head of a linked list, determine if the linked list has a cycle in it.

There is a cycle in a linked list if there is some node in the list that can be reached again by continuously following the `next` pointer. Internally, `pos` is used to denote the index of the node that tail's `next` pointer is connected to. **Note that `pos` is not passed as a parameter.**

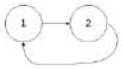
Return `true` *if there is a cycle in the linked list*. Otherwise, return `false`.

Example 1:




```
Input: head = [3,2,0,-4], pos = 1
Output: true
Explanation: There is a cycle in the linked list, where the tail connects to the 1st node (0-indexed).
```

Example 2:




```
Input: head = [1,2], pos = 0
Output: true
Explanation: There is a cycle in the linked list, where the tail connects to the 0th node.
```

Example 3:


```
Input: head = [1], pos = -1
```

17.1K | 465 | 171 Online

Code

C++ | Auto

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    bool hasCycle(ListNode *head) {
        if(head==NULL) return 0;

        struct ListNode *slow=head;
        struct ListNode *fast=head;

        while(fast!=NULL && fast->next!=NULL)
        {
            slow=slow->next;
            fast=fast->next->next;

            if(slow==fast)
                return 1;
        }
        return 0;
    }
};
```

Saved | Ln 17, Col 40

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

head =

[3,2,0,-4]

pos =

Activate Windows
Go to Settings to activate Windows.

23°C Mostly sunny | Search | ENG IN | 10:58 24-11-2025

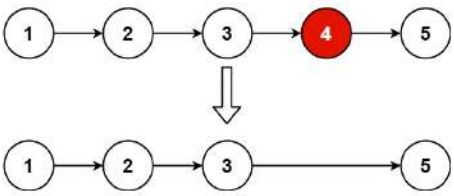
19. Remove Nth Node From End of List

Solved

Medium · Topics · Companies · Hint

Given the `head` of a linked list, remove the `n`th node from the end of the list and return its head.

Example 1:




```
Input: head = [1,2,3,4,5], n = 2
Output: [1,2,3,5]
```

Example 2:

```
Input: head = [1], n = 1
Output: []
```

Example 3:

```
Input: head = [1,2], n = 1
Output: [1]
```

Constraints:

- The number of nodes in the list is `sz`.

```c
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
struct ListNode* removeNthFromEnd(struct ListNode* head, int n) {
    int count=0;
        struct ListNode *ptr=head;
        while(ptr!=NULL)
        {
            ptr=ptr->next;
            count++;
        }
        if(n==count)
        {
            struct ListNode *newHead = head->next;
            free(head);
            return newHead;
        }
        struct ListNode *temp=head;
        struct ListNode *prev;
        for(int j=1;j<(count-n);j++)
            temp=temp->next;
        prev=temp->next;
        temp->next=prev->next;
        free(prev);
        return head;
}
```

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

head =

[1,2,3,4,5]

Palindrome Linked List - LeetCo | New tab

leetcode.com/problems/palindrome-linked-list/submissions/1838689717/

Problem List | Submit | Premium

Description | Editorial | Solutions | Submissions

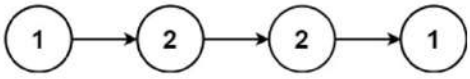
234. Palindrome Linked List

Solved

Easy | Topics | Companies

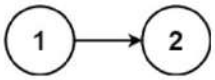
Given the `head` of a singly linked list, return `true` *if it is a palindrome* or `false` *otherwise.*

Example 1:




```
Input: head = [1,2,2,1]
Output: true
```

Example 2:




```
Input: head = [1,2]
Output: false
```

Constraints:

18K | 357 | 127 Online

Code

C | Auto

```
bool isPalindrome(struct ListNode* head) {
    struct ListNode *slow=head,*fast=head,*prev=NULL,
*temp;
    while(fast!=NULL&&fast->next!=NULL){
        fast=fast->next->next;
        temp=slow->next;
        slow->next=prev;
        prev=slow,slow=temp;
    }
    slow=(fast?slow->next:slow);
    while(slow){
        if(slow->val!=prev->val) return false;
        else slow=slow->next,prev=prev->next;
    }
    return true;
```

Saved | Ln 21, Col 5

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2

Input

head =
[1,2,2,1]

Output

true

Leet | Accepted

All Submissions

Accepted 93 / 93 testcases passed

mo... submitted at Nov 24, 2025 22:12

Solution

LIMITED TIME OFFER - $40 off A...

LeetCode's Thanksgiving Sale IS NOW LIVE...

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

44.77 MB Beats 48.79%