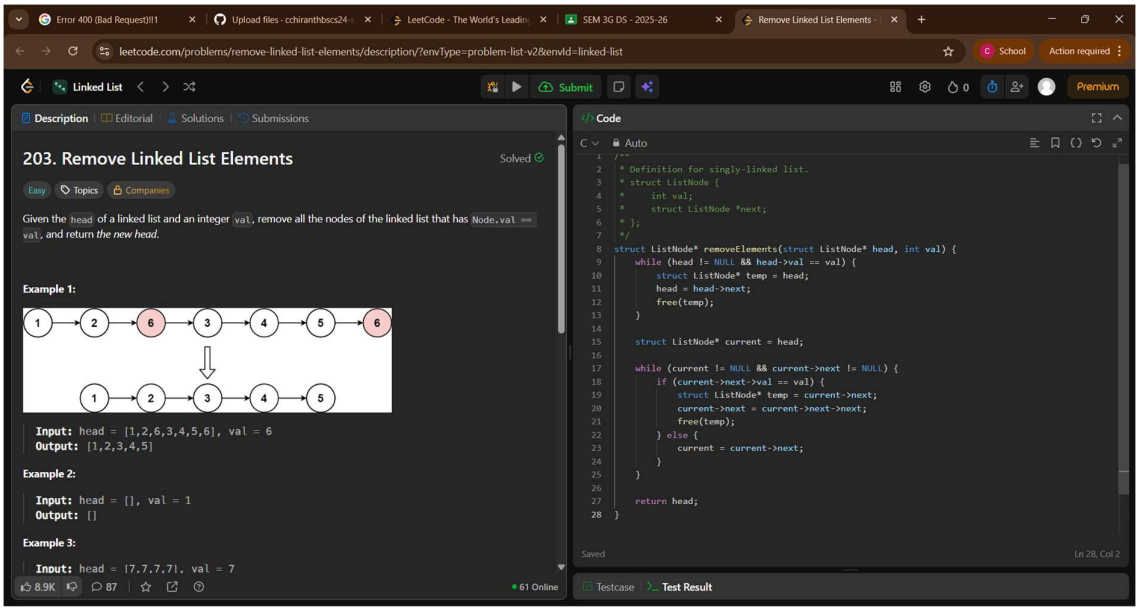
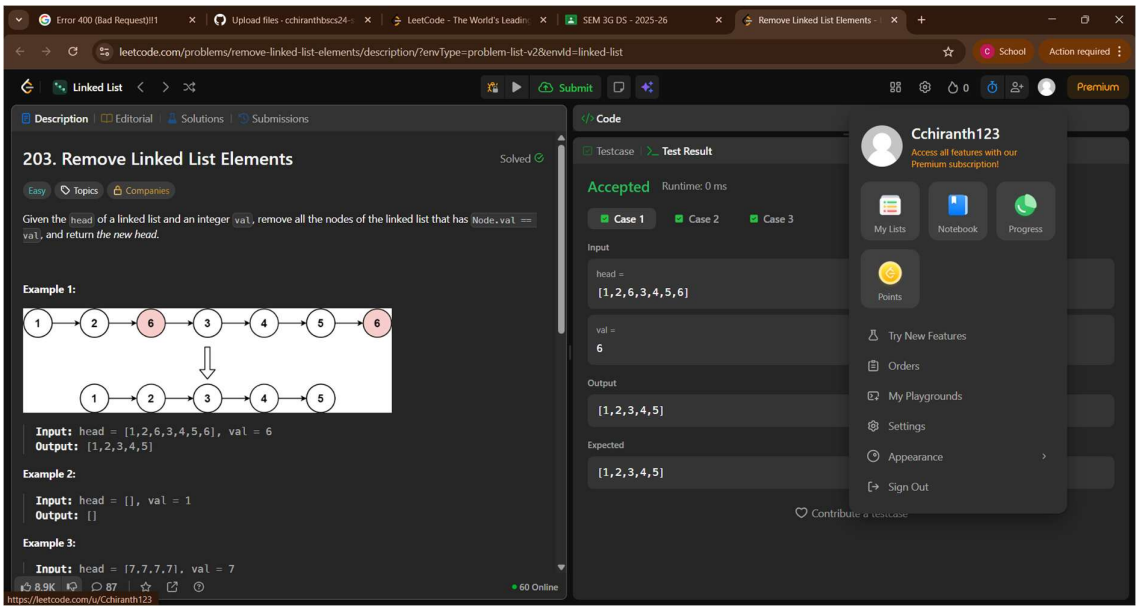

Error 400 (Bad Request)!!1
Upload files - cchiranthbscs24-
LeetCode - The World's Leadin
SEM 3G DS - 2025-26
Remove Linked List Elements -
leetcode.com/problems/remove-linked-list-elements/description/?envType=problem-list-v2&envId=linked-list
School
Action required
Linked List
Submit
Premium
Description
Editorial
Solutions
Submissions
203. Remove Linked List Elements
Solved
Easy
Topics
Companies
Given the head of a linked list and an integer val, remove all the nodes of the linked list that has Node.val == val, and return the new head.
Example 1:
Input: head = [1,2,6,3,4,5,6], val = 6
Output: [1,2,3,4,5]
Example 2:
Input: head = [], val = 1
Output: []
Example 3:
Input: head = [7,7,7,7], val = 7
8.9K
87
61 Online
Code
C
Auto
/**
* Definition for singly-linked list.
* struct ListNode {
* int val;
* struct ListNode *next;
* };
*/
struct ListNode* removeElements(struct ListNode* head, int val) {
while (head != NULL && head->val == val) {
struct ListNode* temp = head;
head = head->next;
free(temp);
}
struct ListNode* current = head;
while (current != NULL && current->next != NULL) {
if (current->next->val == val) {
struct ListNode* temp = current->next;
current->next = current->next->next;
free(temp);
} else {
current = current->next;
}
}
return head;
}
Saved
Ln 28, Col 2
Testcase
Test Result


Error 400 (Bad Request)!!1
Upload files - cchiranthbscs24-
LeetCode - The World's Leadin
SEM 3G DS - 2025-26
Remove Linked List Elements -
leetcode.com/problems/remove-linked-list-elements/description/?envType=problem-list-v2&envId=linked-list
School
Action required
Linked List
Submit
Premium
Description
Editorial
Solutions
Submissions
203. Remove Linked List Elements
Solved
Easy
Topics
Companies
Given the head of a linked list and an integer val, remove all the nodes of the linked list that has Node.val == val, and return the new head.
Example 1:
Input: head = [1,2,6,3,4,5,6], val = 6
Output: [1,2,3,4,5]
Example 2:
Input: head = [], val = 1
Output: []
Example 3:
Input: head = [7,7,7,7], val = 7
8.9K
87
60 Online
https://leetcode.com/u/Cchiranth123
Code
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
head =
[1,2,6,3,4,5,6]
val =
6
Output
[1,2,3,4,5]
Expected
[1,2,3,4,5]
Cchiranth123
Access all features with our Premium subscription!
My Lists
Notebook
Progress
Points
Try New Features
Orders
My Playgrounds
Settings
Appearance
Sign Out
Contribute a testcase

Error 400 (Bad Request)!!1 | Upload files - cchiranthbscs24 | LeetCode - The World's Leading | SEM 3G DS - 2025-26 | Sort List - LeetCode

leetcode.com/problems/sort-list/description/?envType=problem-list-v2&envId=linked-list

School | Action required

Linked List | Submit | Premium

Description | Editorial | Solutions | Submissions

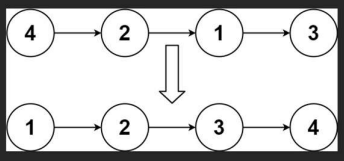
148. Sort List

Solved

Medium | Topics | Companies

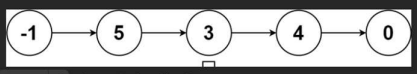
Given the `head` of a linked list, return *the list after sorting it in* ***ascending order***.

Example 1:



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

Example 2:



12.8K | 162 | 109 Online

Code

C | Auto

```c
struct ListNode* merge(struct ListNode* l1, struct ListNode* l2) {
    if (l1 == NULL) return l2;
    if (l2 == NULL) return l1;

    struct ListNode* head = NULL;
    struct ListNode* tail = NULL;

    while (l1 && l2) {
        struct ListNode* temp = NULL;
        if (l1->val < l2->val) {
            temp = l1;
            l1 = l1->next;
        } else {
            temp = l2;
            l2 = l2->next;
        }

        if (!head) {
            head = tail = temp;
        } else {
            tail->next = temp;
            tail = temp;
        }
    }

    if (l1) tail->next = l1;
    if (l2) tail->next = l2;

```

Saved | Ln 60, Col 1

Testcase | Test Result

Error 400 (Bad Request)!!1 | Upload files - cchiranthbscs24 | LeetCode - The World's Leading | SEM 3G DS - 2025-26 | Sort List - LeetCode

leetcode.com/problems/sort-list/description/?envType=problem-list-v2&envId=linked-list

School | Action required

Linked List | Submit | Premium

Description | Editorial | Solutions | Submissions

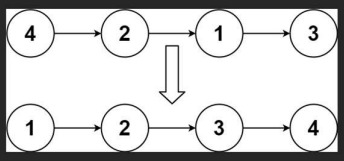
148. Sort List

Solved

Medium | Topics | Companies

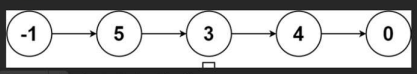
Given the head of a linked list, return the list after sorting it in ascending order.

Example 1:



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

Example 2:



12.8K | 162 | 110 Online

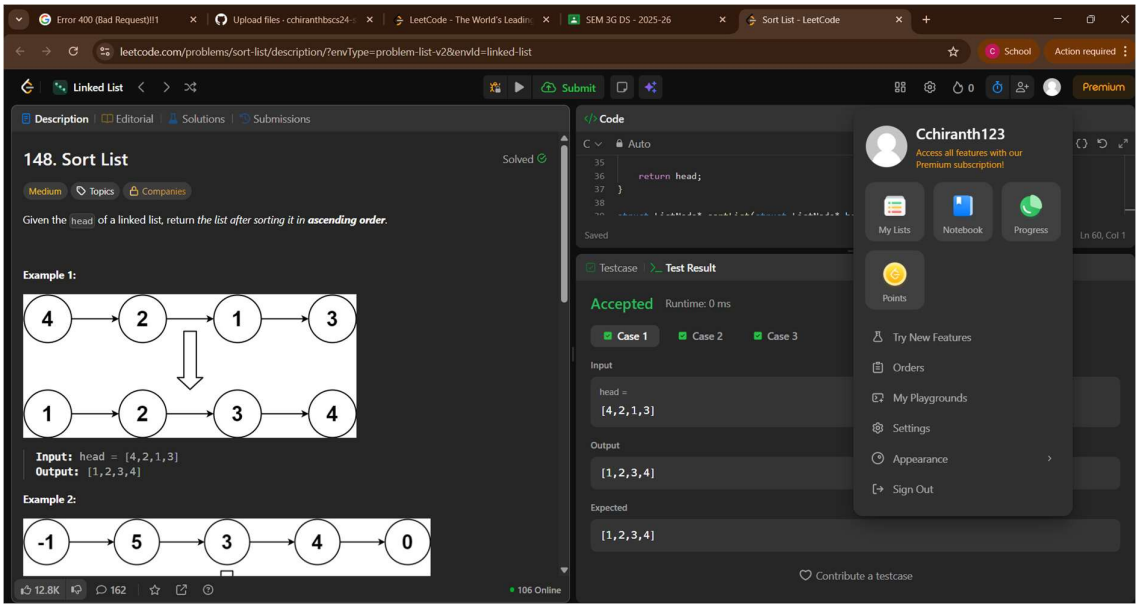
Code

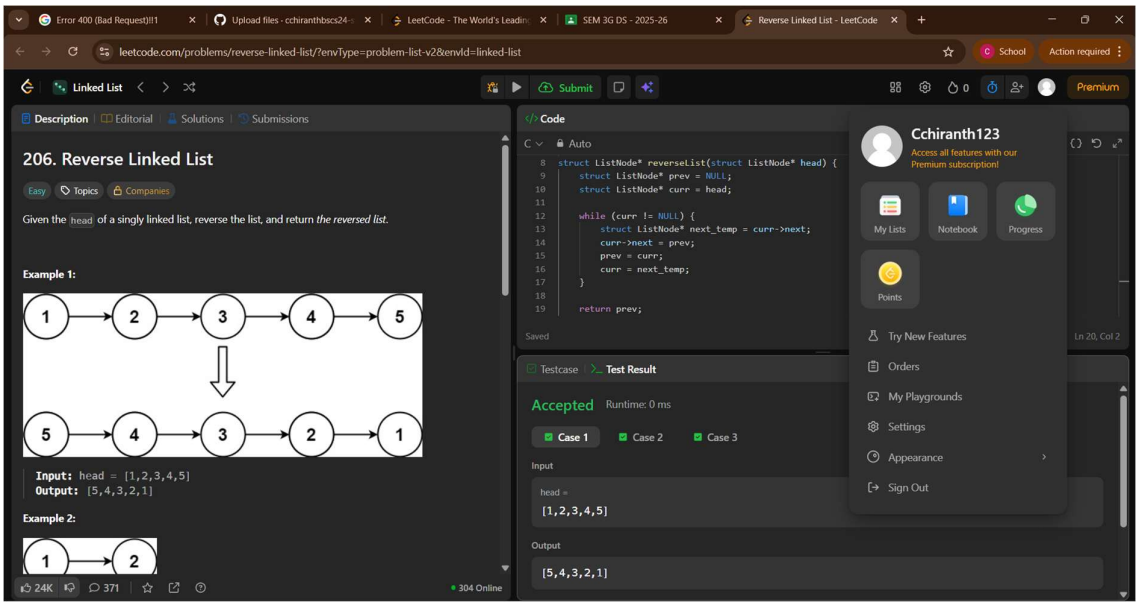
C | Auto

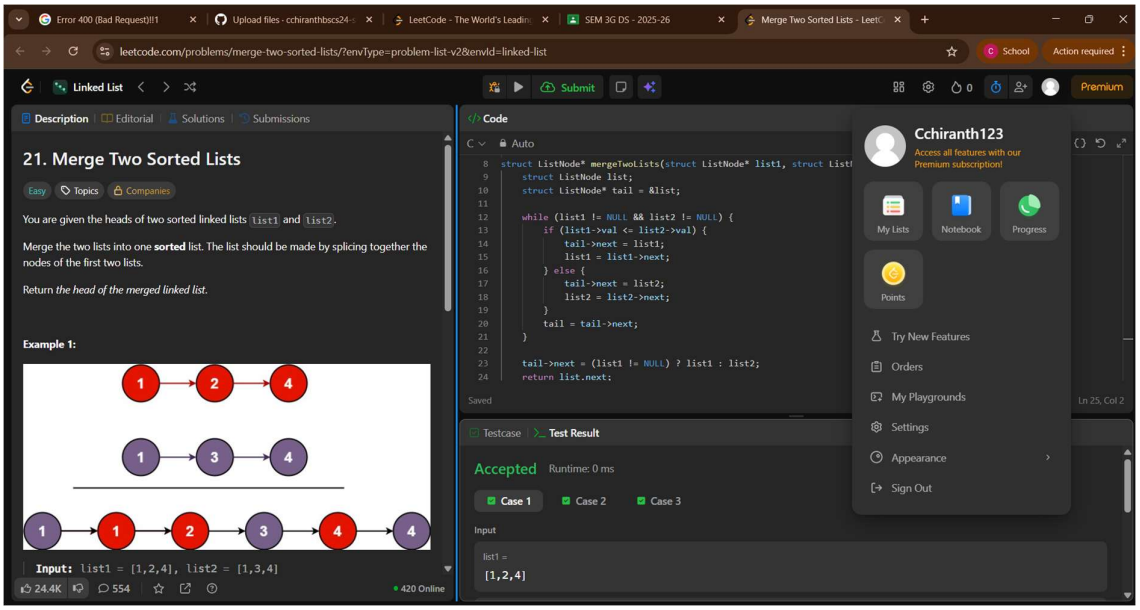
```c

    return head;
}

struct ListNode* sortList(struct ListNode* head) {
    if (head == NULL || head->next == NULL)
        return head;

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

    while (fast && fast->next) {
        prev = slow;
        slow = slow->next;
        fast = fast->next->next;
    }

    prev->next = NULL;

    struct ListNode* left = sortList(head);
    struct ListNode* right = sortList(slow);

    return merge(left, right);
}

```

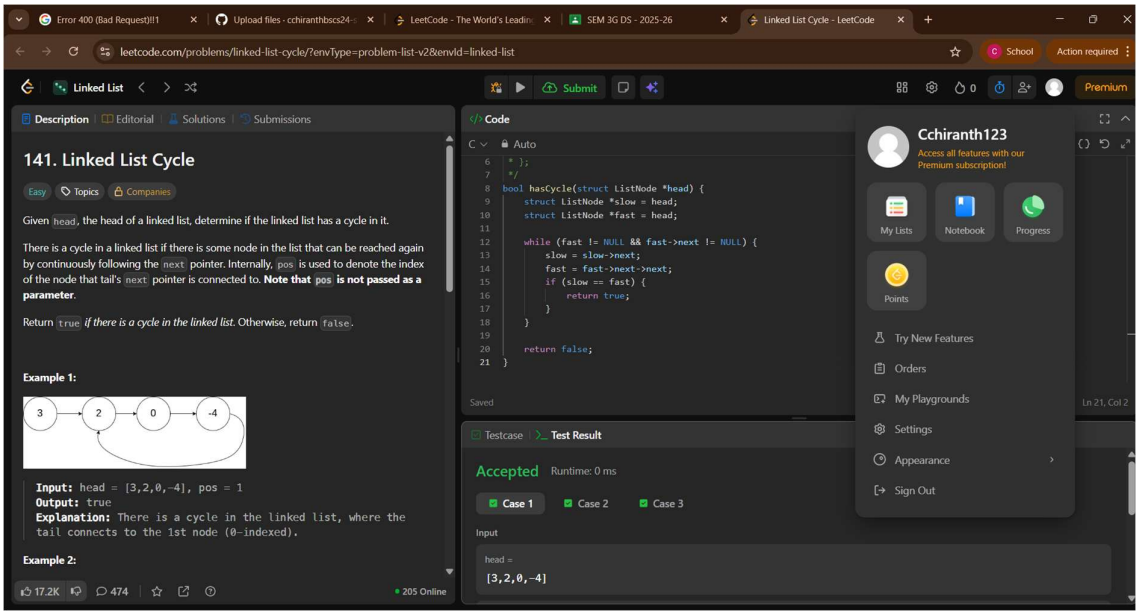
Saved | Ln 60, Col 1

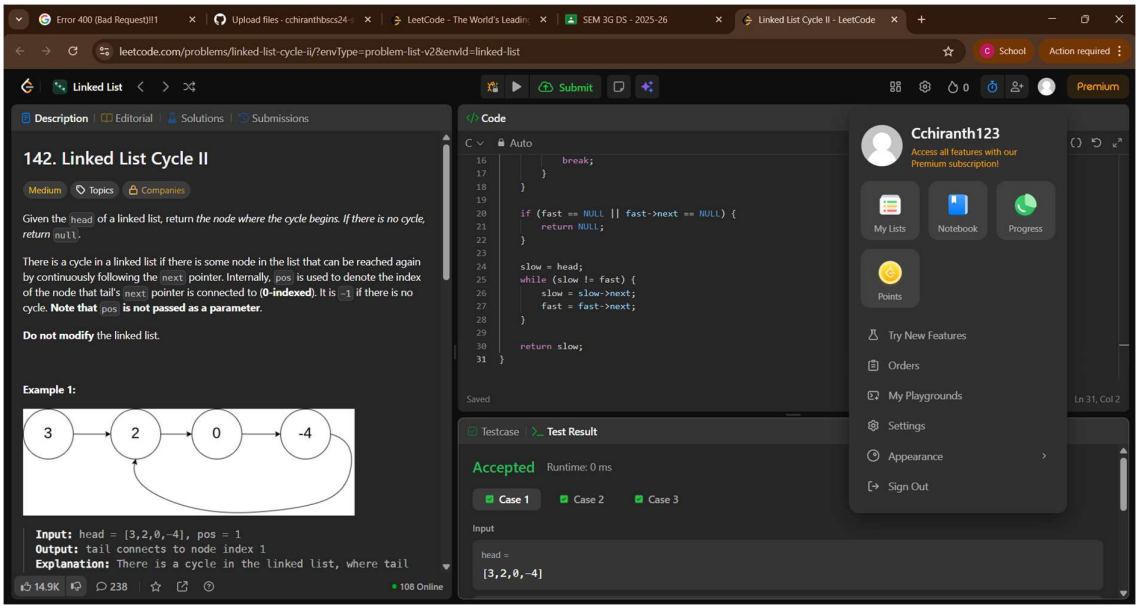
Testcase | Test Result


leetcode.com/problems/sort-list/description/?envType=problem-list-v2&envId=linked-list
Linked List
Submit
Premium
Description | Editorial | Solutions | Submissions
148. Sort List
Solved
Medium
Topics
Companies
Given the head of a linked list, return the list after sorting it in ascending order.
Example 1:
Input: head = [4,2,1,3]
Output: [1,2,3,4]
Example 2:
12.8K
162
106 Online
Code
C
Auto
return head;
Saved
Ln 60, Col 1
Testcase | Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
head =
[4,2,1,3]
Output
[1,2,3,4]
Expected
[1,2,3,4]
Contribute a testcase
Cchiranth123
Access all features with our Premium subscription!
My Lists
Notebook
Progress
Points
Try New Features
Orders
My Playgrounds
Settings
Appearance
Sign Out


leetcode.com/problems/reverse-linked-list/?envType=problem-list-v2&envId=linked-list
Linked List
Submit
Premium
Description | Editorial | Solutions | Submissions
206. Reverse Linked List
Easy
Topics
Companies
Given the head of a singly linked list, reverse the list, and return the reversed list.
Example 1:
Input: head = [1,2,3,4,5]
Output: [5,4,3,2,1]
Example 2:
24K
371
304 Online
Code
C
Auto
struct ListNode* reverseList(struct ListNode* head) {
struct ListNode* prev = NULL;
struct ListNode* curr = head;
while (curr != NULL) {
struct ListNode* next_temp = curr->next;
curr->next = prev;
prev = curr;
curr = next_temp;
}
return prev;
Saved
Ln 20, Col 2
Testcase | Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
head =
[1,2,3,4,5]
Output
[5,4,3,2,1]
Cchiranth123
Access all features with our Premium subscription!
My Lists
Notebook
Progress
Points
Try New Features
Orders
My Playgrounds
Settings
Appearance
Sign Out


leetcode.com/problems/merge-two-sorted-lists/?envType=problem-list-v2&envId=linked-list
Linked List
Submit
Premium
Description
Editorial
Solutions
Submissions
21. Merge Two Sorted Lists
Easy
Topics
Companies
You are given the heads of two sorted linked lists list1 and list2.
Merge the two lists into one sorted list. The list should be made by splicing together the nodes of the first two lists.
Return the head of the merged linked list.
Example 1:
Input: list1 = [1,2,4], list2 = [1,3,4]
24.4K
554
420 Online
Code
C
Auto
struct ListNode* mergeTwoLists(struct ListNode* list1, struct ListNode* list2) {
struct ListNode list;
struct ListNode* tail = &list;
while (list1 != NULL && list2 != NULL) {
if (list1->val <= list2->val) {
tail->next = list1;
list1 = list1->next;
} else {
tail->next = list2;
list2 = list2->next;
}
tail = tail->next;
}
tail->next = (list1 != NULL) ? list1 : list2;
return list.next;
Saved
Ln 25, Col 2
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
list1 =
[1,2,4]
Cchiranth123
Access all features with our Premium subscription!
My Lists
Notebook
Progress
Points
Try New Features
Orders
My Playgrounds
Settings
Appearance
Sign Out


leetcode.com/problems/linked-list-cycle/?envType=problem-list-v2&envId=linked-list
Linked List
Submit
Premium
Description
Editorial
Solutions
Submissions
141. Linked List Cycle
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:
17.2K
474
205 Online
Code
C
Auto
* };
*/
bool hasCycle(struct ListNode *head) {
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 true;
}
}
return false;
}
Saved
Ln 21, Col 2
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
head =
[3,2,0,-4]
Cchiranth123
Access all features with our Premium subscription!
My Lists
Notebook
Progress
Points
Try New Features
Orders
My Playgrounds
Settings
Appearance
Sign Out


Error 400 (Bad Request)!!1
Upload files - cchiranthbscs24-
LeetCode - The World's Leadin
SEM 3G DS - 2025-26
Linked List Cycle II - LeetCode
leetcode.com/problems/linked-list-cycle-ii/?envType=problem-list-v2&envId=linked-list
School
Action required
Linked List
Submit
Premium
Description
Editorial
Solutions
Submissions
142. Linked List Cycle II
Medium
Topics
Companies
Given the head of a linked list, return the node where the cycle begins. If there is no cycle, return null.
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 (0-indexed). It is -1 if there is no cycle. Note that pos is not passed as a parameter.
Do not modify the linked list.
Example 1:
3
2
0
-4
Input: head = [3,2,0,-4], pos = 1
Output: tail connects to node index 1
Explanation: There is a cycle in the linked list, where tail
14.9K
238
108 Online
Code
C
Auto
break;
}
}
if (fast == NULL || fast->next == NULL) {
return NULL;
}
slow = head;
while (slow != fast) {
slow = slow->next;
fast = fast->next;
}
return slow;
}
Saved
Ln 31, Col 2
Cchiranth123
Access all features with our Premium subscription!
My Lists
Notebook
Progress
Points
Try New Features
Orders
My Playgrounds
Settings
Appearance
Sign Out
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
head =
[3,2,0,-4]