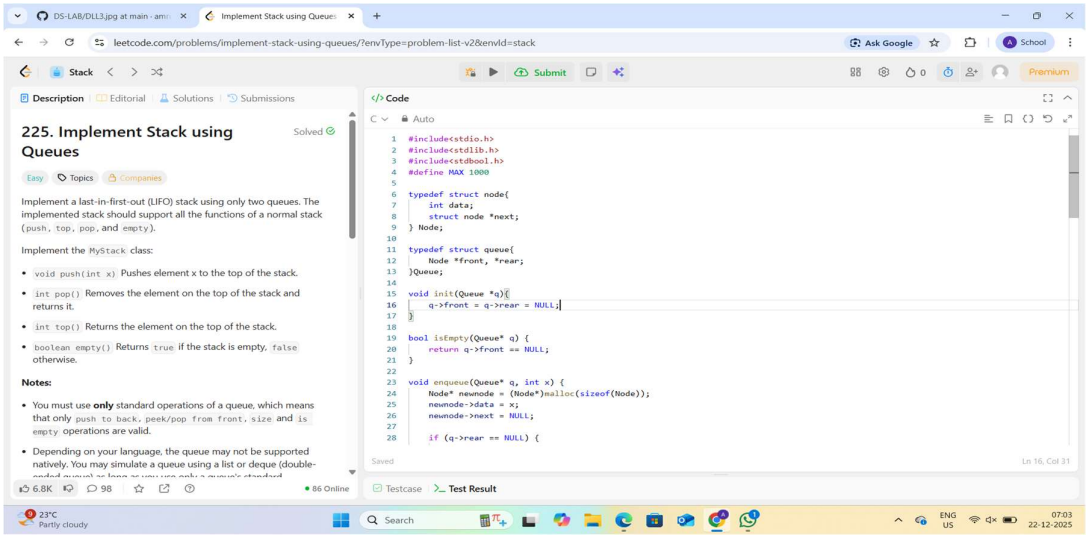
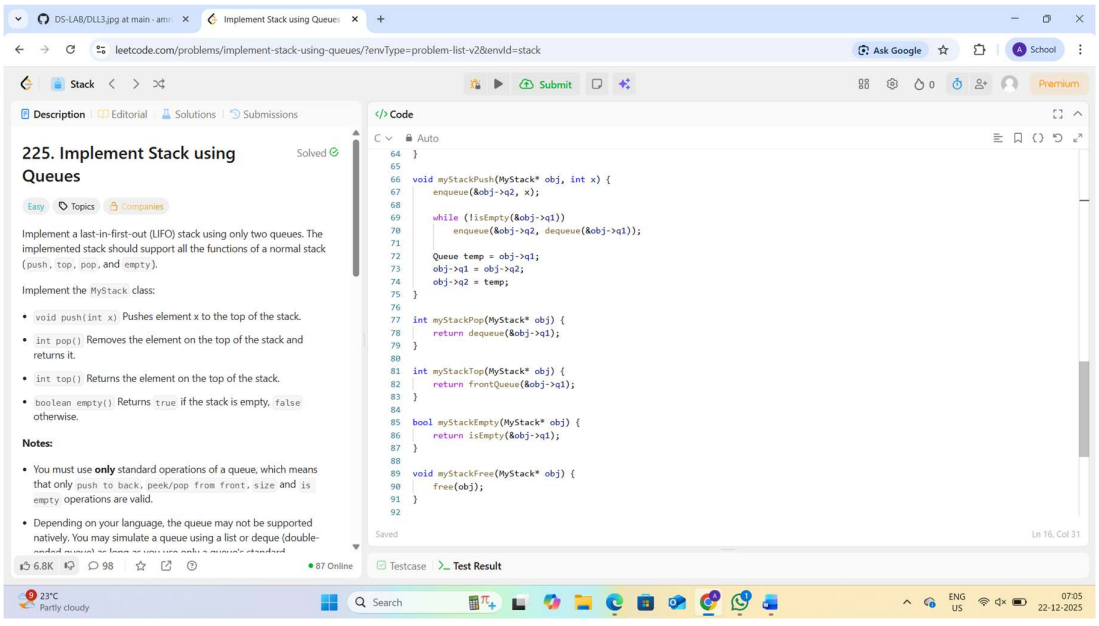
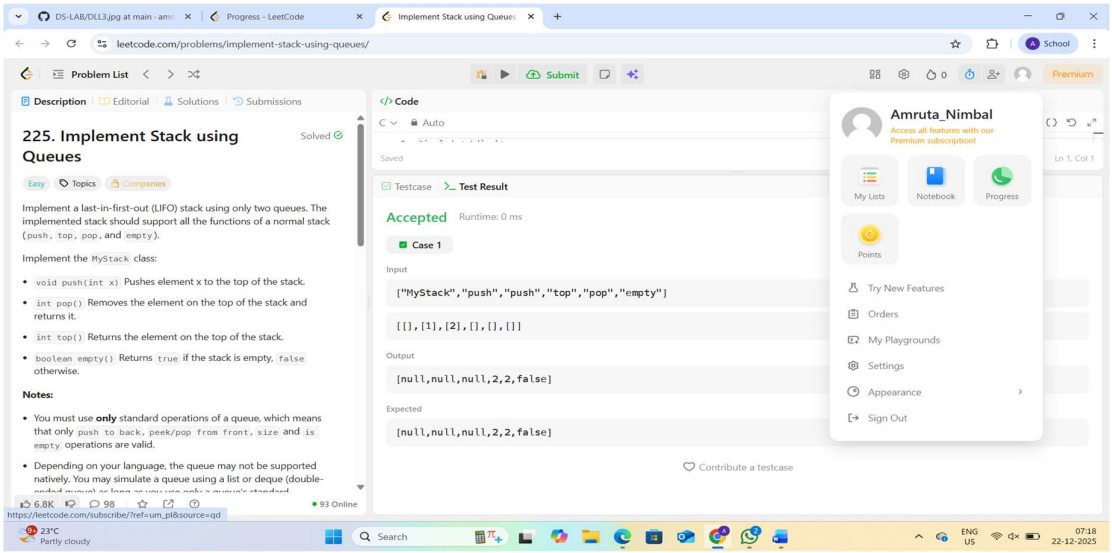
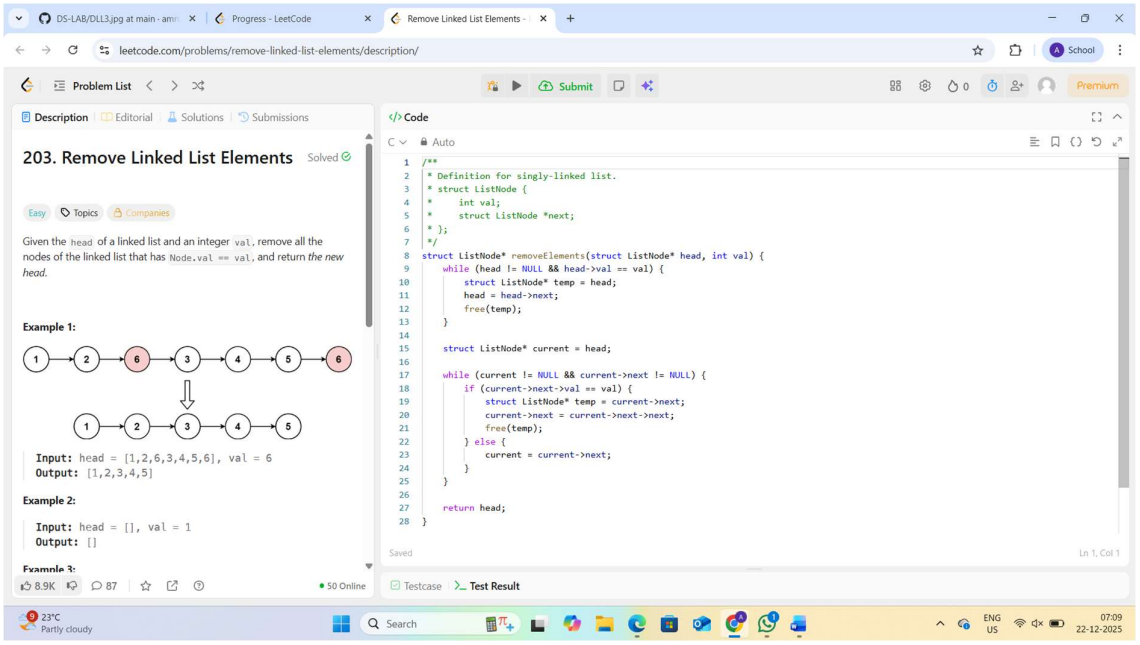
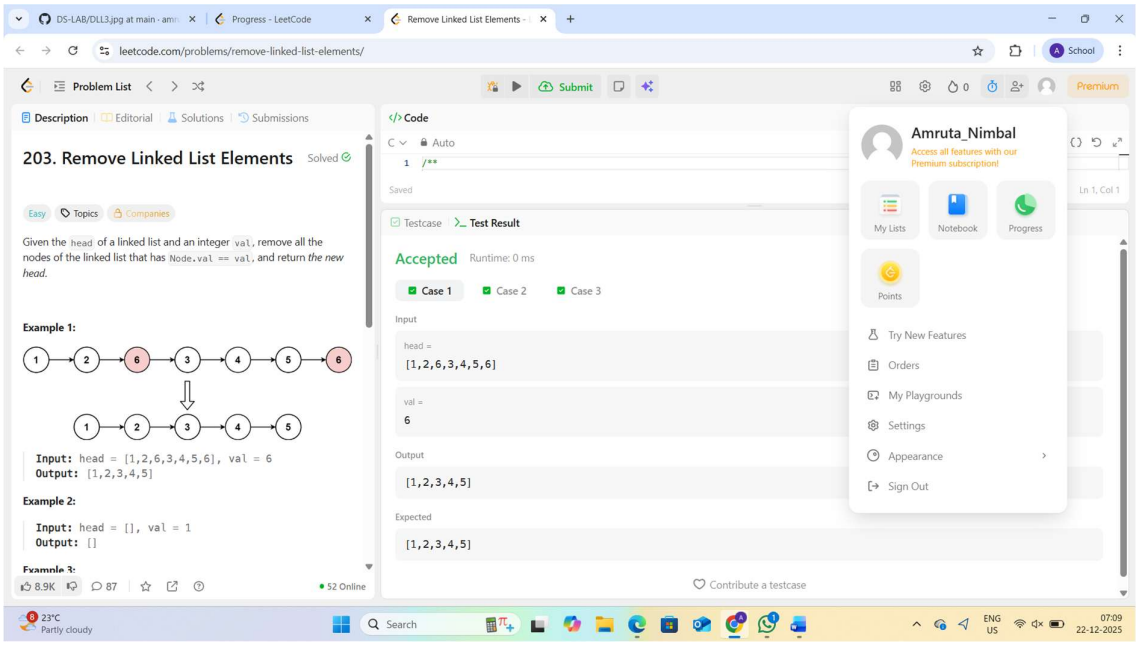
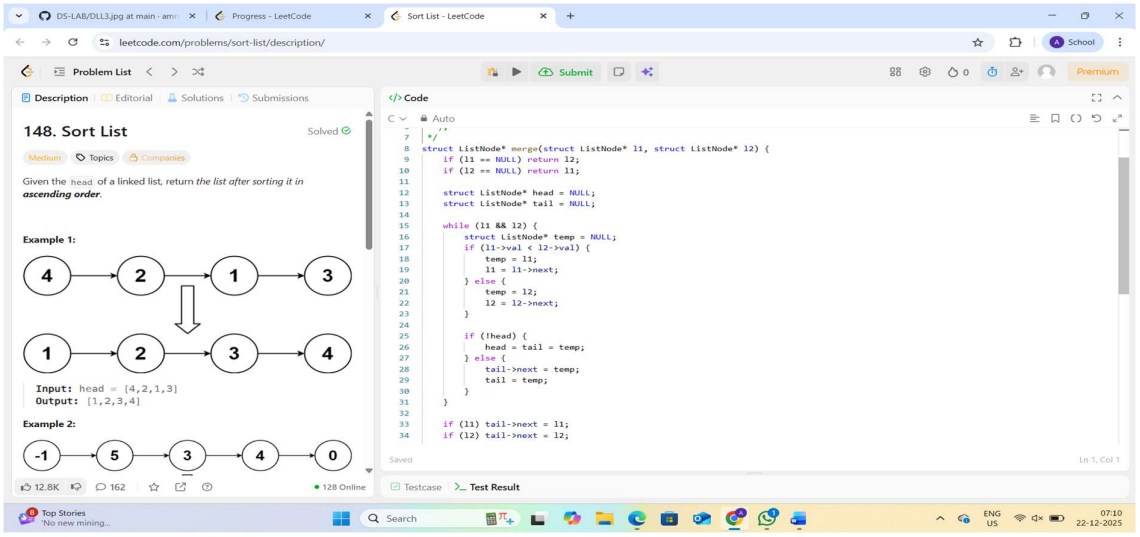

Implement Stack using Queues
leetcode.com/problems/implement-stack-using-queues/?envType=problem-list-v2&envId=stack
Stack
Submit
Premium
Description
Editorial
Solutions
Submissions
225. Implement Stack using Queues
Solved
Easy
Topics
Companies
Implement a last-in-first-out (LIFO) stack using only two queues. The implemented stack should support all the functions of a normal stack (push, top, pop, and empty).
Implement the MyStack class:
void push(int x) Pushes element x to the top of the stack.
int pop() Removes the element on the top of the stack and returns it.
int top() Returns the element on the top of the stack.
boolean empty() Returns true if the stack is empty, false otherwise.
Notes:
You must use only standard operations of a queue, which means that only push to back, peek/pop from front, size and is empty operations are valid.
Depending on your language, the queue may not be supported natively. You may simulate a queue using a list or deque (double-
Code
C
Auto
#include<stdio.h>
#include<stdlib.h>
#include<stdbool.h>
#define MAX 1000
typedef struct node{
int data;
struct node *next;
} Node;
typedef struct queue{
Node *front, *rear;
}Queue;
void init(Queue *q){
q->front = q->rear = NULL;
}
bool isEmpty(Queue* q) {
return q->front == NULL;
}
void enqueue(Queue* q, int x) {
Node* newnode = (Node*)malloc(sizeof(Node));
newnode->data = x;
newnode->next = NULL;
if (q->rear == NULL) {
Saved
Ln 16, Col 31
Testcase
Test Result
6.8K
98
86 Online
23°C
Partly cloudy
Search
ENG
US
07:03
22-12-2025

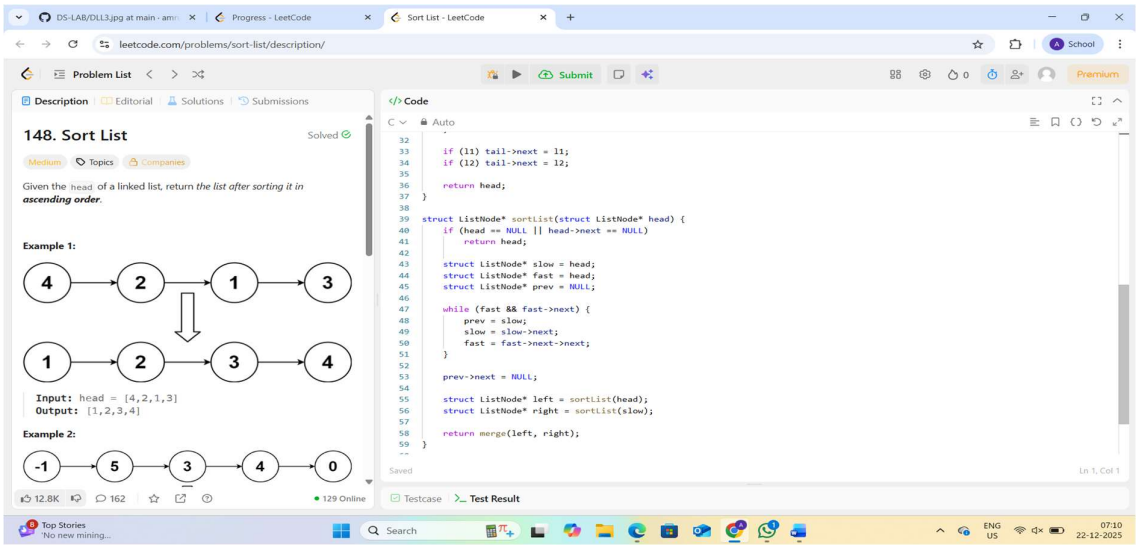

Implement Stack using Queues
leetcode.com/problems/implement-stack-using-queues/?envType=problem-list-v2&envId=stack
Stack
Submit
Premium
Description
Editorial
Solutions
Submissions
225. Implement Stack using Queues
Solved
Easy
Topics
Companies
Implement a last-in-first-out (LIFO) stack using only two queues. The implemented stack should support all the functions of a normal stack (push, top, pop, and empty).
Implement the MyStack class:
void push(int x) Pushes element x to the top of the stack.
int pop() Removes the element on the top of the stack and returns it.
int top() Returns the element on the top of the stack.
boolean empty() Returns true if the stack is empty, false otherwise.
Notes:
You must use only standard operations of a queue, which means that only push to back, peek/pop from front, size and is empty operations are valid.
Depending on your language, the queue may not be supported natively. You may simulate a queue using a list or deque (double-
Code
C
Auto
}
void myStackPush(MyStack* obj, int x) {
enqueue(&obj->q2, x);
while (!isEmpty(&obj->q1))
enqueue(&obj->q2, dequeue(&obj->q1));
Queue temp = obj->q1;
obj->q1 = obj->q2;
obj->q2 = temp;
}
int myStackPop(MyStack* obj) {
return dequeue(&obj->q1);
}
int myStackTop(MyStack* obj) {
return frontQueue(&obj->q1);
}
bool myStackEmpty(MyStack* obj) {
return isEmpty(&obj->q1);
}
void myStackFree(MyStack* obj) {
free(obj);
}
Saved
Ln 16, Col 31
Testcase
Test Result
6.8K
98
87 Online
23°C
Partly cloudy
Search
ENG
US
07:05
22-12-2025

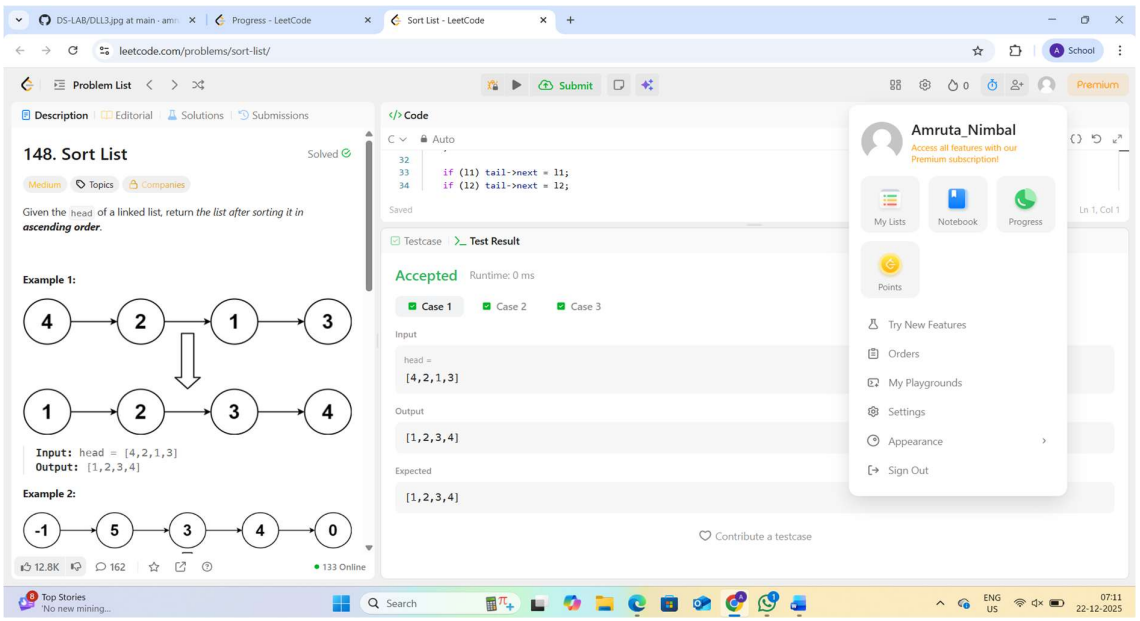

Progress - LeetCode
Implement Stack using Queues
leetcode.com/problems/implement-stack-using-queues/
Problem List
Submit
Premium
Description
Editorial
Solutions
Submissions
225. Implement Stack using Queues
Solved
Easy
Topics
Companies
Implement a last-in-first-out (LIFO) stack using only two queues. The implemented stack should support all the functions of a normal stack (push, top, pop, and empty).
Implement the MyStack class:
void push(int x) Pushes element x to the top of the stack.
int pop() Removes the element on the top of the stack and returns it.
int top() Returns the element on the top of the stack.
boolean empty() Returns true if the stack is empty, false otherwise.
Notes:
You must use only standard operations of a queue, which means that only push to back, peek/pop from front, size and is empty operations are valid.
Depending on your language, the queue may not be supported natively. You may simulate a queue using a list or deque (double-
Code
C
Auto
Saved
Ln 1, Col 1
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Input
["MyStack","push","push","top","pop","empty"]
[[],[1],[2],[],[],[]]
Output
[null,null,null,2,2,false]
Expected
[null,null,null,2,2,false]
Contribute a testcase
Amruta_Nimbal
Access all features with our Premium subscription!
My Lists
Notebook
Progress
Points
Try New Features
Orders
My Playgrounds
Settings
Appearance
Sign Out
6.8K
98
93 Online
https://leetcode.com/subscribe/?ref=um_pl&source=qd
23°C
Partly cloudy
Search
ENG
US
07:18
22-12-2025

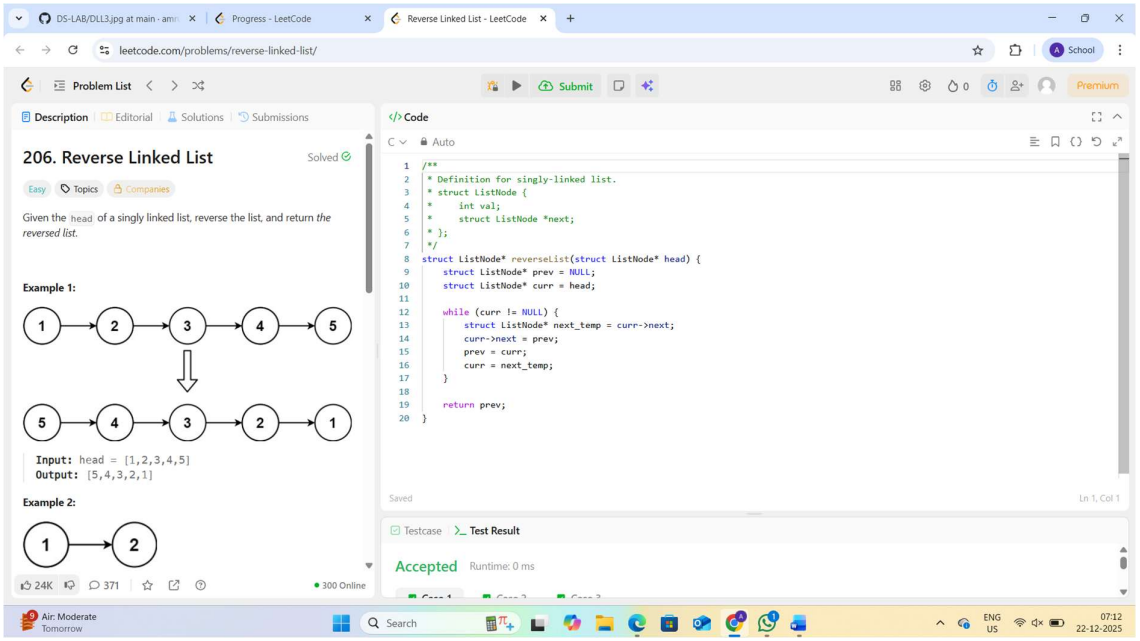

DS-LAB/DLL3.jpg at main · amr
Progress - LeetCode
Remove Linked List Elements -
leetcode.com/problems/remove-linked-list-elements/description/
School
Problem List
Submit
Premium
Description
Editorial
Solutions
Submissions
203. Remove Linked List Elements
Solved
Easy
Topics
Companies
Given the head of 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:
8.9K
87
50 Online
Code
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 1, Col 1
Testcase
Test Result
23°C
Partly cloudy
Search
ENG
US
07:09
22-12-2025

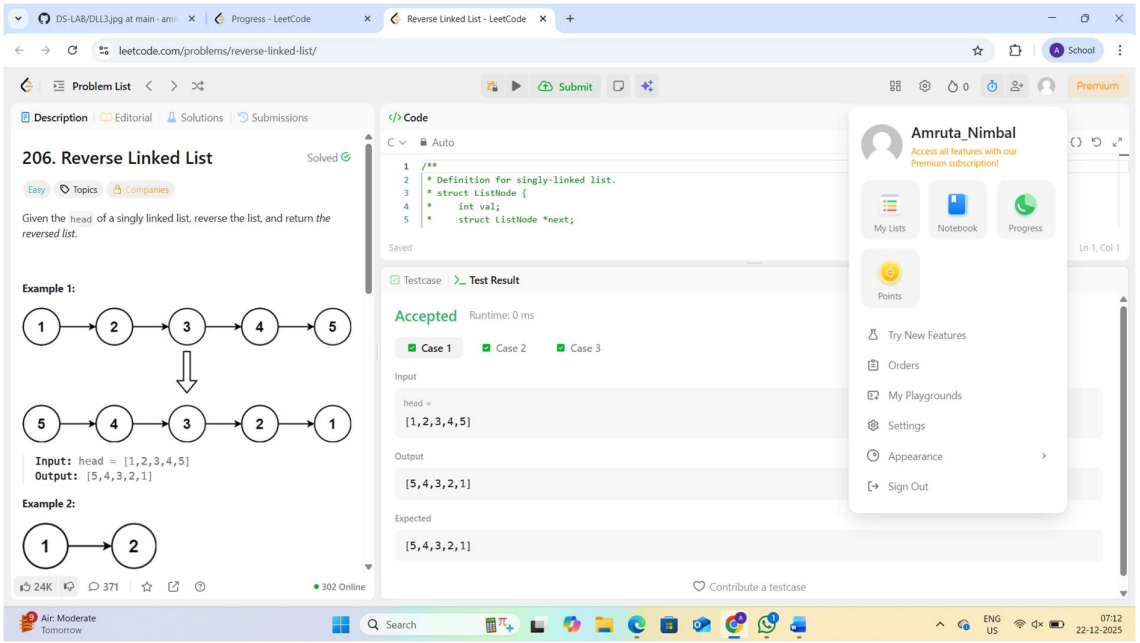

DS-LAB/DLL3.jpg at main · amr
Progress - LeetCode
Remove Linked List Elements -
leetcode.com/problems/remove-linked-list-elements/
School
Problem List
Submit
Premium
Description
Editorial
Solutions
Submissions
203. Remove Linked List Elements
Solved
Easy
Topics
Companies
Given the head of a linked list 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:
8.9K
87
52 Online
Code
Auto
/**
Saved
Ln 1, Col 1
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]
Contribute a testcase
Amruta_Nimbal
Access all features with our Premium subscription!
My Lists
Notebook
Progress
Points
Try New Features
Orders
My Playgrounds
Settings
Appearance
Sign Out
23°C
Partly cloudy
Search
ENG
US
07:09
22-12-2025

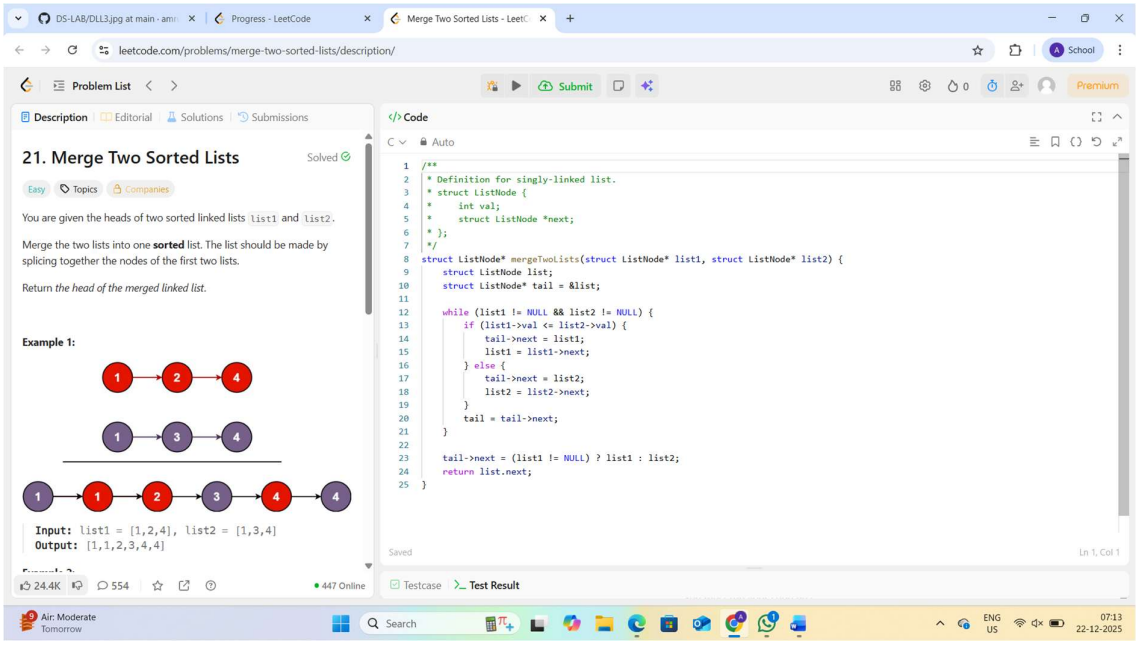

DS-LAB/DLL3.jpg at main · amr
Progress - LeetCode
Sort List - LeetCode
leetcode.com/problems/sort-list/description/
Problem List
Submit
Premium
Description
Editorial
Solutions
Submissions
148. Sort List
Solved
Medium
Topics
Companies
Given the head of a linked list, return the list after sorting it in ascending order.
Example 1:
Input: head = [4,2,1,3]
Output: [1,2,3,4]
Example 2:
12.8K
162
128 Online
Code
Auto
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 1, Col 1
Testcase
Test Result
Top Stories
No new mining...
Search
ENG
US
07:10
22-12-2025

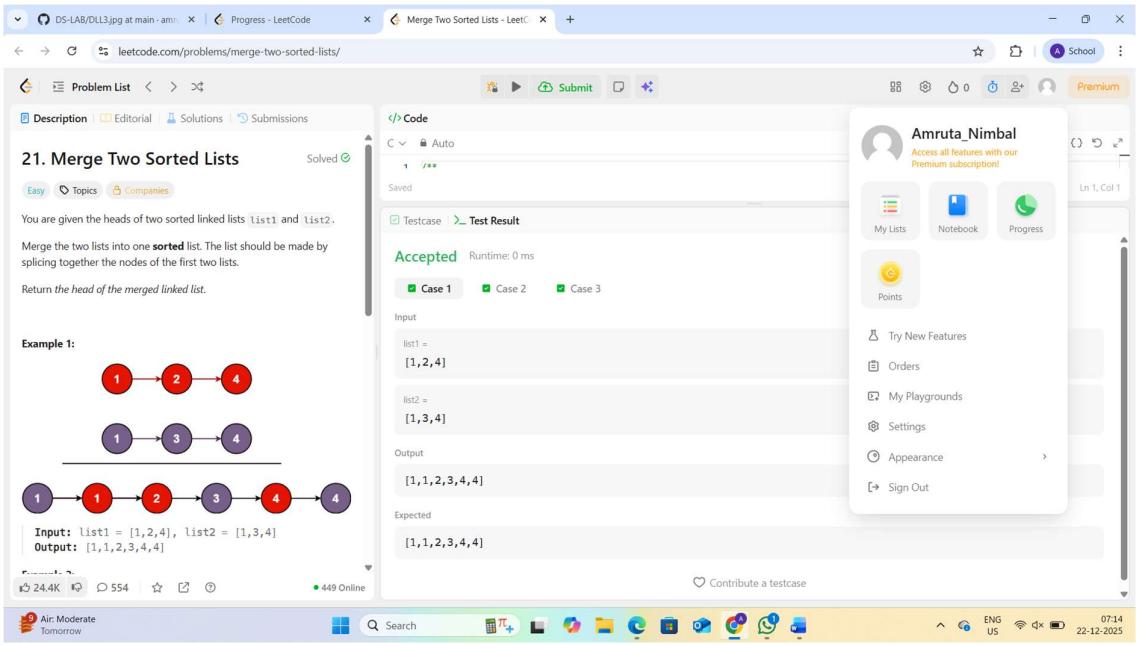

DS-LAB/DLL3.jpg at main · amr
Progress - LeetCode
Sort List - LeetCode
leetcode.com/problems/sort-list/description/
Problem List
Submit
Premium
Description
Editorial
Solutions
Submissions
148. Sort List
Solved
Medium
Topics
Companies
Given the head of a linked list, return the list after sorting it in ascending order.
Example 1:
Input: head = [4,2,1,3]
Output: [1,2,3,4]
Example 2:
12.8K
162
129 Online
Code
Auto
if (l1) tail->next = l1;
if (l2) tail->next = l2;
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 1, Col 1
Testcase
Test Result
Top Stories
No new mining...
Search
ENG
US
07:10
22-12-2025

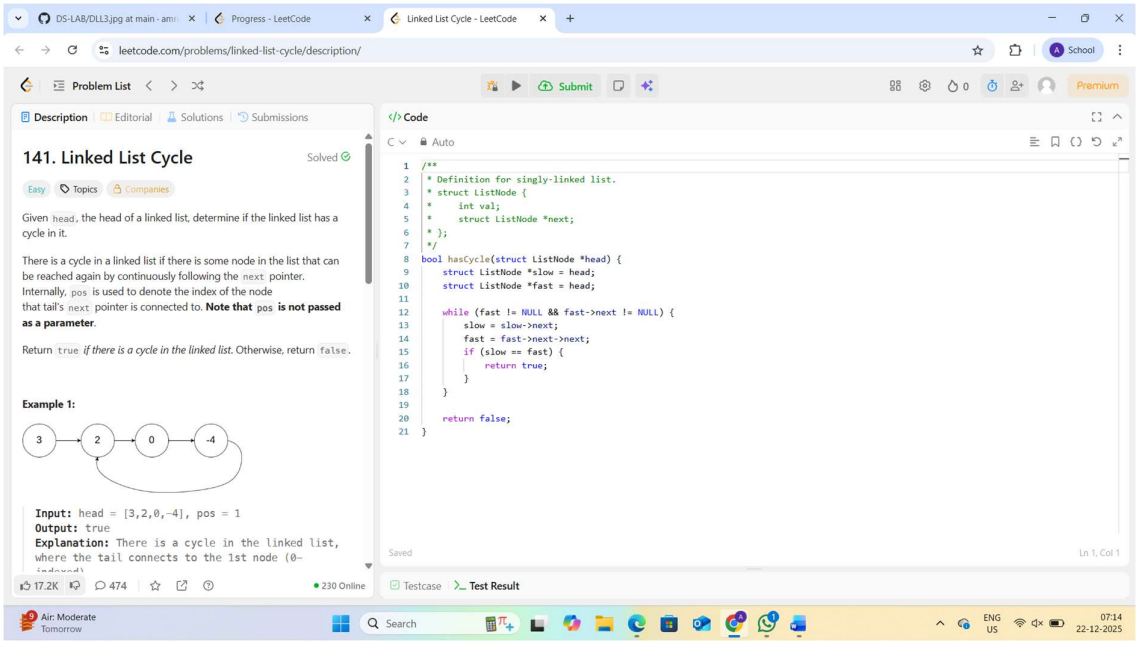

DS-LAB/DLL3.jpg at main · amr
Progress - LeetCode
Sort List - LeetCode
leetcode.com/problems/sort-list/
Problem List
Submit
Premium
Description
Editorial
Solutions
Submissions
148. Sort List
Solved
Medium
Topics
Companies
Given the head of a linked list, return the list after sorting it in ascending order.
Example 1:
Input: head = [4,2,1,3]
Output: [1,2,3,4]
Example 2:
12.8K
162
133 Online
Code
Auto
if (l1) tail->next = l1;
if (l2) tail->next = l2;
Saved
Ln 1, 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
Amruta_Nimbal
Access all features with our Premium subscription!
My Lists
Notebook
Progress
Points
Try New Features
Orders
My Playgrounds
Settings
Appearance
Sign Out
Top Stories
No new mining...
Search
ENG
US
07:11
22-12-2025

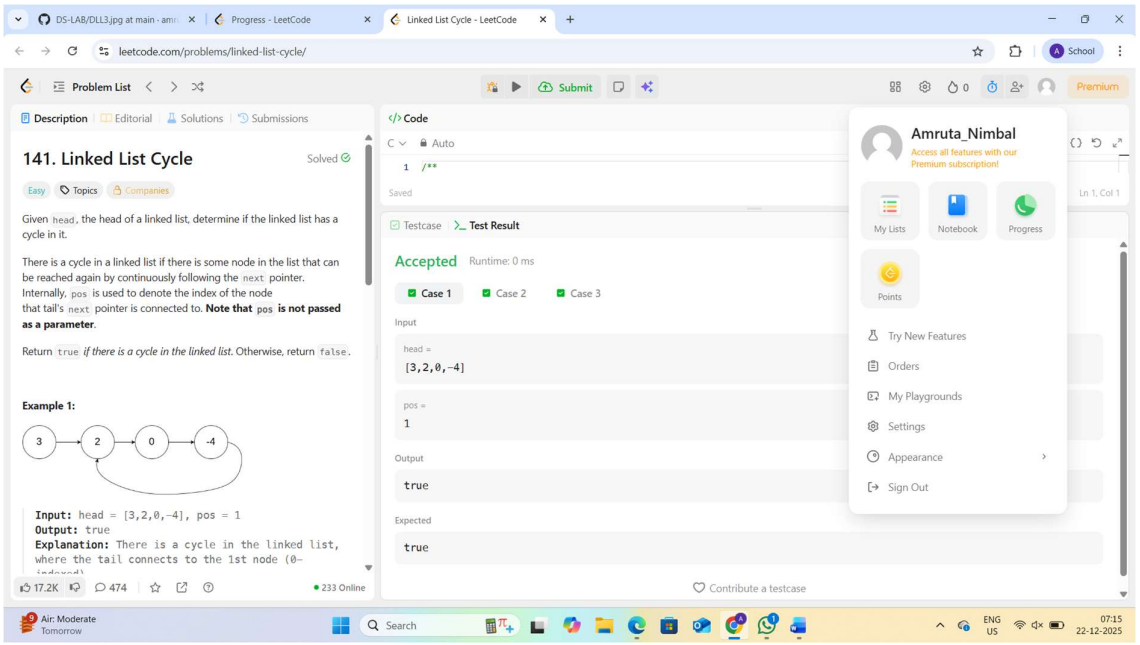

Reverse Linked List - LeetCode
leetcode.com/problems/reverse-linked-list/
Problem List
Submit
Premium
Description
Editorial
Solutions
Submissions
206. Reverse Linked List
Solved
Easy
Topics
Companies
Given the head of a singly linked list, reverse the list, and return the reversed list.
Example 1:
Input: head = [1,2,3,4,5]
Output: [5,4,3,2,1]
Example 2:
24K
371
300 Online
Code
Auto
/**
* Definition for singly-linked list.
* struct ListNode {
*   int val;
*   struct ListNode *next;
* };
*/
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 1, Col 1
Testcase
Test Result
Accepted
Runtime: 0 ms
Air: Moderate
Tomorrow
Search
ENG
US
07:12
22-12-2025


Reverse Linked List - LeetCode
leetcode.com/problems/reverse-linked-list/
Problem List
Submit
Premium
Description
Editorial
Solutions
Submissions
206. Reverse Linked List
Solved
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
302 Online
Code
Auto
/**
* Definition for singly-linked list.
* struct ListNode {
*   int val;
*   struct ListNode *next;
Saved
Ln 1, Col 1
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]
Expected
[5,4,3,2,1]
Contribute a testcase
Amruta_Nimbal
Access all features with our Premium subscription!
My Lists
Notebook
Progress
Points
Try New Features
Orders
My Playgrounds
Settings
Appearance
Sign Out
Air: Moderate
Tomorrow
Search
ENG
US
07:12
22-12-2025


DS-LAB/DLL3.jpg at main · amr
Progress - LeetCode
Merge Two Sorted Lists - Leet
leetcode.com/problems/merge-two-sorted-lists/description/
Problem List
Submit
Premium
Description
Editorial
Solutions
Submissions
21. Merge Two Sorted Lists
Solved
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]
Output: [1,1,2,3,4,4]
24.4K
554
447 Online
Code
Auto
/**
* Definition for singly-linked list.
* struct ListNode {
* int val;
* struct ListNode *next;
* };
*/
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 1, Col 1
Testcase
Test Result
Air: Moderate
Tomorrow
Search
ENG
US
07:13
22-12-2025


DS-LAB/DLL3.jpg at main · amr
Progress - LeetCode
Merge Two Sorted Lists - Leet
leetcode.com/problems/merge-two-sorted-lists/
Problem List
Submit
Premium
Description
Editorial
Solutions
Submissions
21. Merge Two Sorted Lists
Solved
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]
Output: [1,1,2,3,4,4]
24.4K
554
449 Online
Code
Auto
Saved
Ln 1, Col 1
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
list1 =
[1,2,4]
list2 =
[1,3,4]
Output
[1,1,2,3,4,4]
Expected
[1,1,2,3,4,4]
Contribute a testcase
Amruta_Nimbal
Access all features with our Premium subscription!
My Lists
Notebook
Progress
Points
Try New Features
Orders
My Playgrounds
Settings
Appearance
Sign Out
Air: Moderate
Tomorrow
Search
ENG
US
07:14
22-12-2025


Linked List Cycle - LeetCode
leetcode.com/problems/linked-list-cycle/description/
Problem List
Submit
Premium
Description
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:
3
2
0
-4
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-
17.2K
474
230 Online
Code
Auto
/**
* Definition for singly-linked list.
* struct ListNode {
* int val;
* struct ListNode *next;
* };
*/
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 1, Col 1
Testcase
Test Result
Air: Moderate
Tomorrow
Search
ENG
US
07:14
22-12-2025


Linked List Cycle - LeetCode
leetcode.com/problems/linked-list-cycle/
Problem List
Submit
Premium
Description
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:
3
2
0
-4
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-
17.2K
474
233 Online
Code
Auto
/**
Saved
Ln 1, Col 1
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
head =
[3,2,0,-4]
pos =
1
Output
true
Expected
true
Contribute a testcase
Amruta_Nimbal
Access all features with our Premium subscription!
My Lists
Notebook
Progress
Points
Try New Features
Orders
My Playgrounds
Settings
Appearance
Sign Out
Air: Moderate
Tomorrow
Search
ENG
US
07:15
22-12-2025

DS-LAB/DLL3.jpg at main · amr | Progress - LeetCode | Linked List Cycle II - LeetCode

leetcode.com/problems/linked-list-cycle-ii/description/

Problem List | Submit | Premium

Description | Editorial | Solutions | Submissions

142. Linked List Cycle II

Solved

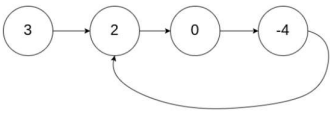
Medium | Topics | Companies

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:




```
Input: head = [3,2,0,-4], pos = 1
Output: tail connects to node index 1
```

14.9K | 237 | 114 Online

Code

Auto

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
struct ListNode *detectCycle(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) {
            break;
        }
    }

    if (fast == NULL || fast->next == NULL) {
        return NULL;
    }

    slow = head;
    while (slow != fast) {
        slow = slow->next;
        fast = fast->next;
    }

    return slow;
}
```

Saved | Ln 1, Col 1

Testcase | Test Result

Trending videos: Kids have the be...

Search

ENG US | 07:16 22-12-2025

DS-LAB/DLL3.jpg at main · amr | Progress - LeetCode | Linked List Cycle II - LeetCode

leetcode.com/problems/linked-list-cycle-ii/

Problem List | Submit | Premium

Description | Editorial | Solutions | Submissions

142. Linked List Cycle II

Solved

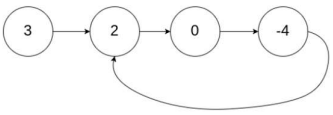
Medium | Topics | Companies

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:




```
Input: head = [3,2,0,-4], pos = 1
Output: tail connects to node index 1
```

14.9K | 237 | 112 Online

Code

Auto

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
```

Saved | Ln 1, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

head =
[3,2,0,-4]

pos =
1

Output

```
tail connects to node index 1
```

Expected

```
tail connects to node index 1
```

Contribute a testcase

Amruta_Nimbal

Access all features with our Premium subscription!

My Lists | Notebook | Progress | Points

Try New Features

Orders

My Playgrounds

Settings

Appearance

Sign Out

Trending videos: Kids have the be...

Search

ENG US | 07:16 22-12-2025