CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

Subject: Advance programming lab

Subject code: 22CSH-351

Assignment No.: 2

Student Name: Muskan

UID: 22BCS13343

Branch: BE-CSE

Section/Group: 22BCS_IOT-607(B)

Submitted to: Er. Pratima sonalie

Date of Submission: 04-03-25

Open-End | Inbox (6,8 | marksheet | Welcome t | Chandigar | Print Linke | New Doc (| Screensho

geeksforgeeks.org/problems/print-linked-list-elements/0

90% Refund

Courses | Tutorials | Jobs | Practice | Contests

Problem | Editorial | Submissions | Comments

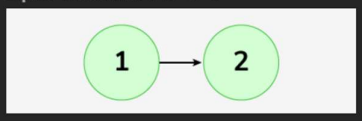
Print Linked List

Difficulty: **Basic** Accuracy: **60.71%** Submissions: **142K+** Points: **1**

Given a linked list. Print all the elements of the linked list separated by space followed.

Examples:

Input: LinkedList : 1 -> 2



Output: 1 2

Explanation: The linked list contains two elements 1 and 2.The elements are printed in a single line.

Input: Linked List : 49 -> 10 -> 30

C++ (g++ 5.4) | Start Timer

```
/*
struct Node {
    int data;
    struct Node* next;

    Node(int x) {
        data = x;
        next = nullptr;
    }
};
*/
/*
    Print elements of a linked list on console
    Head pointer input could be NULL as well for empty list
*/

class Solution {
  public:
    // Function to display the elements of a linked list in same line
    void printList(Node *head) {
        while (head != nullptr) {
        cout << head->data << " ";
        head = head->next;
    }
        //22BCS13343 MUSKAN
    }
};
// } Driver Code Ends
```

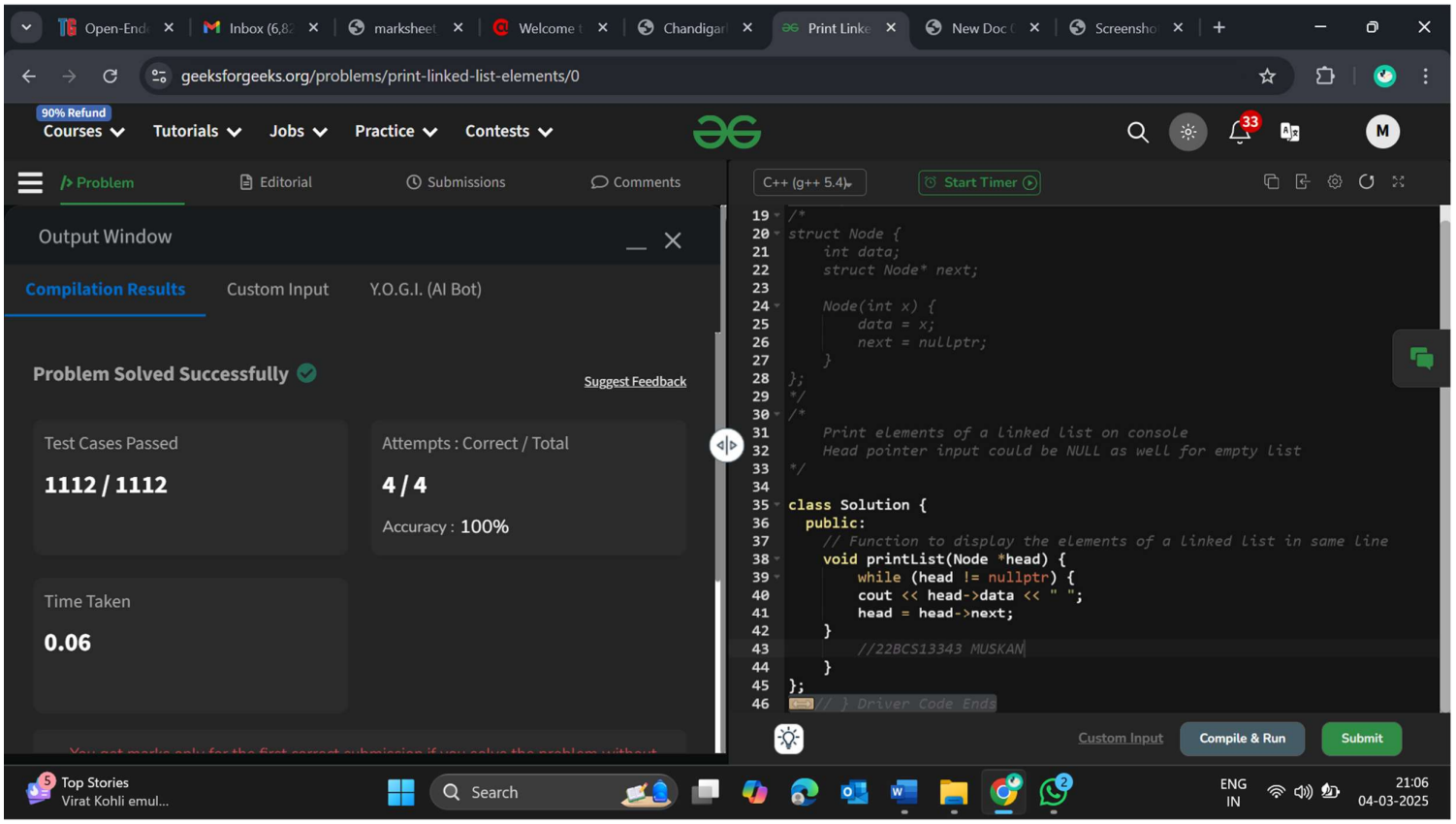
Custom Input | Compile & Run | Submit

Top Stories
Virat Kohli emul...

Search

ENG
IN

21:05
04-03-2025


Open-End
Inbox (6,8
markshee
Welcome t
Chandigar
Print Linke
New Doc (
Screensho
geeksforgeeks.org/problems/print-linked-list-elements/0
90% Refund
Courses
Tutorials
Jobs
Practice
Contests
33
M
Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1112 / 1112
Attempts : Correct / Total
4 / 4
Accuracy : 100%
Time Taken
0.06
C++ (g++ 5.4)
Start Timer
/*
struct Node {
int data;
struct Node* next;
Node(int x) {
data = x;
next = nullptr;
}
};
*/
/*
Print elements of a linked list on console
Head pointer input could be NULL as well for empty list
*/
class Solution {
public:
// Function to display the elements of a linked list in same line
void printList(Node *head) {
while (head != nullptr) {
cout << head->data << " ";
head = head->next;
}
//22BCS13343 MUSKAN
}
};
// } Driver Code Ends
Custom Input
Compile & Run
Submit
Top Stories
Virat Kohli emul...
Search
ENG
IN
21:06
04-03-2025

leetcode.com/problems/remove-duplicates-from-sorted-list/

Problem List | Run | Submit | Premium

Description | Editorial | Solutions | Submissions

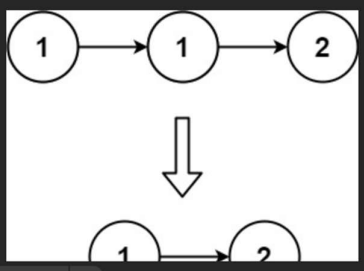
83. Remove Duplicates from Sorted List

Solved

Easy | Topics | Companies

Given the `head` of a sorted linked list, *delete all duplicates such that each element appears only once.* Return *the linked list* ***sorted*** *as well.*

Example 1:



9.1K | 108 | 99 Online

Code | Testcase | Test Result

C++ | Auto

```cpp
class Solution {
public:
    ListNode* deleteDuplicates(ListNode* head) {
        if(head == nullptr || head->next == nullptr) return head;
        ListNode *s1 = head;
        ListNode *s2 = s1->next;
        while( s1 != nullptr && s1->next != nullptr)
        {
            if(s1->val == s2->val)
            {
                s1 -> next = s2 -> next;
                s2 = s1->next;
            }
            else
            {
                s1 = s1->next;
            }
        }
        return head;//22BCS13343 MUSKAN
    }
};
```

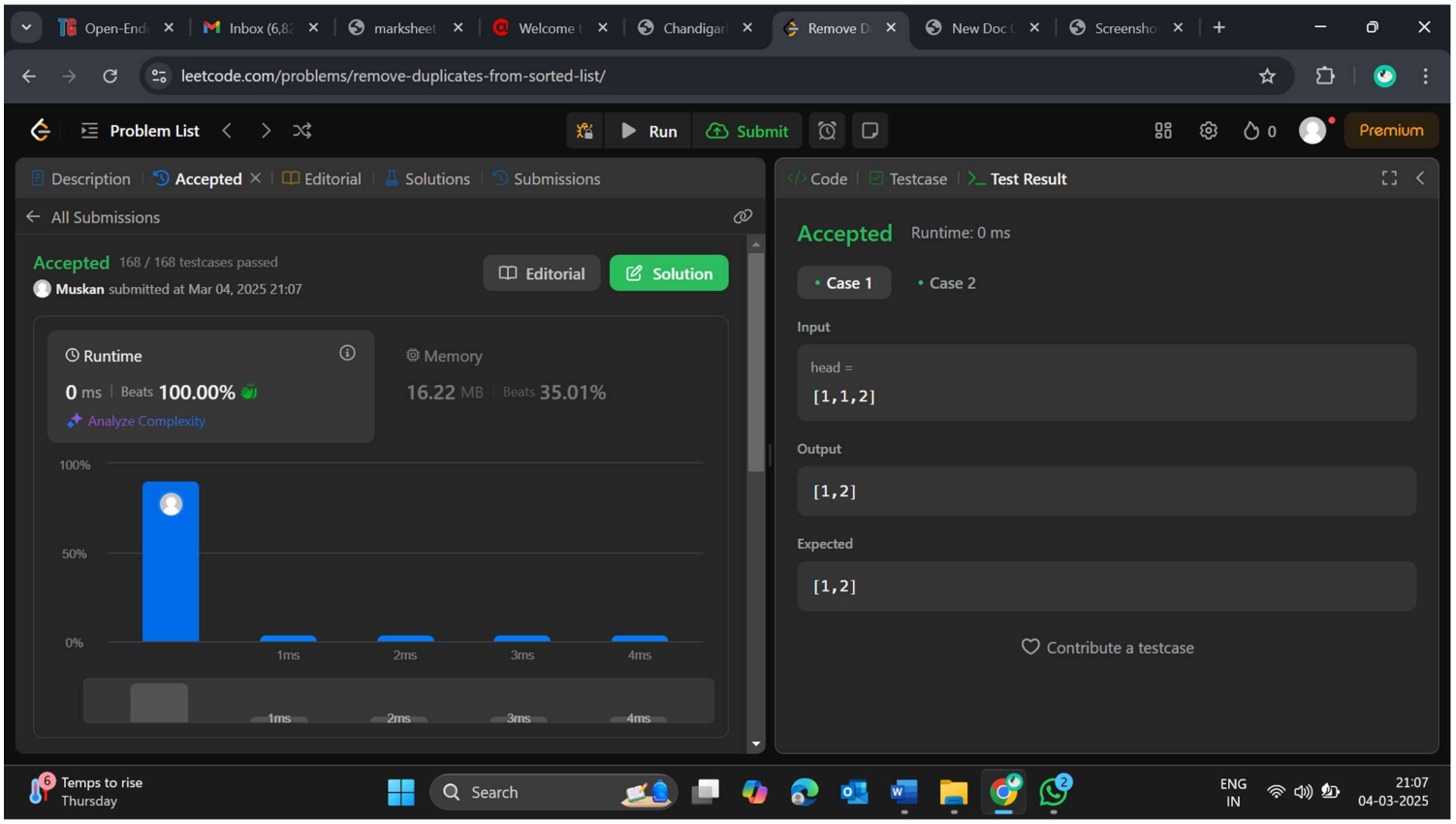
Saved | Ln 29, Col 40

Temps to rise
Thursday

Search

ENG
IN

21:06
04-03-2025


leetcode.com/problems/remove-duplicates-from-sorted-list/
Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 168 / 168 testcases passed
Muskan submitted at Mar 04, 2025 21:07
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
16.22 MB | Beats 35.01%
100%
50%
0%
1ms
2ms
3ms
4ms
Code
Testcase
Test Result
Accepted Runtime: 0 ms
Case 1
Case 2
Input
head =
[1,1,2]
Output
[1,2]
Expected
[1,2]
Contribute a testcase
Temps to rise
Thursday
Search
ENG
IN
21:07
04-03-2025

leetcode.com/problems/reverse-linked-list/description/

Problem List | Run | Submit | Premium

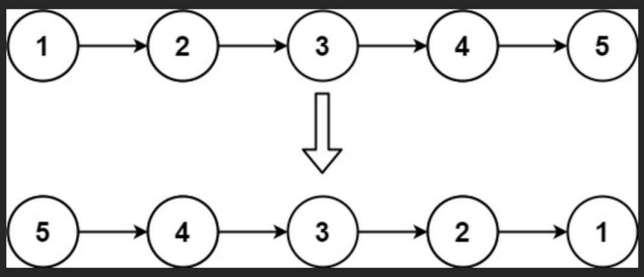
Description | Editorial | Solutions | Submissions

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:



22.6K | 272 | 367 Online

Code | Testcase | Test Result

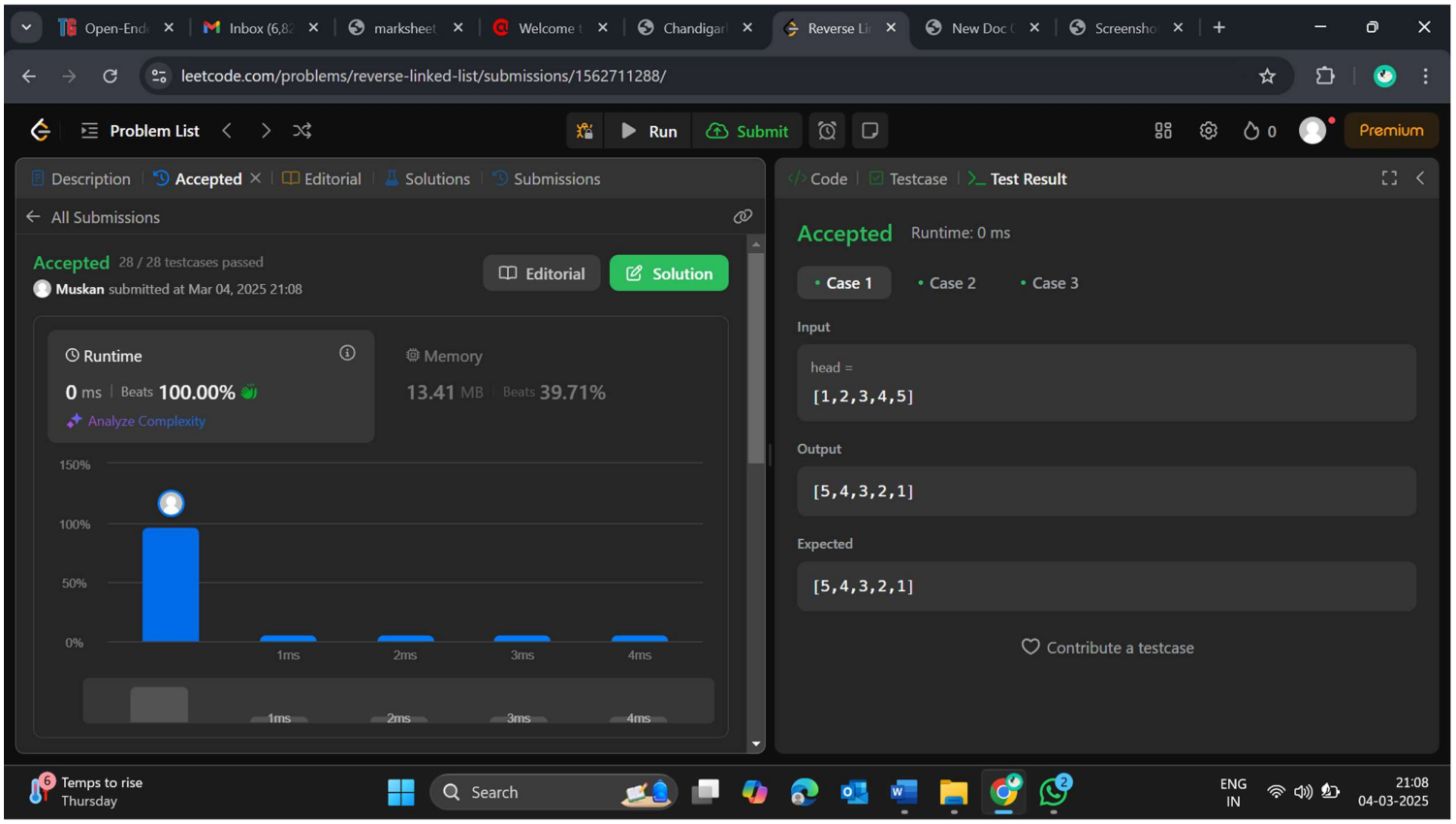
C++ | Auto

```
 *     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:
    ListNode* reverseList(ListNode* head) {
        ListNode* node = nullptr;

        while (head != nullptr) {
            ListNode* temp = head->next;
            head->next = node;
            node = head;
            head = temp;
        }
        //22BCS13343 MUSKAN
        return node;
    }
};
```

Saved | Ln 22, Col 28

Temps to rise Thursday | Search | ENG IN | 21:08 04-03-2025


leetcode.com/problems/reverse-linked-list/submissions/1562711288/
Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 28 / 28 testcases passed
Muskan submitted at Mar 04, 2025 21:08
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
13.41 MB | Beats 39.71%
150%
100%
50%
0%
1ms
2ms
3ms
4ms
Code
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
Temps to rise
Thursday
Search
ENG
IN
21:08
04-03-2025

leetcode.com/problems/delete-the-middle-node-of-a-linked-list/

Problem List | Run | Submit | Premium

Description | Editorial | Solutions | Submissions

2095. Delete the Middle Node of a Linked List

Solved

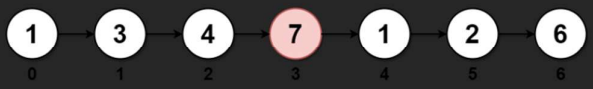
Medium | Topics | Companies | Hint

You are given the `head` of a linked list. **Delete** the **middle node**, and return *the* `head` *of the modified linked list.*

The **middle node** of a linked list of size `n` is the `⌊n / 2⌋`th node from the **start** using **0-based indexing**, where `⌊x⌋` denotes the largest integer less than or equal to `x`.

- For `n` = `1`, `2`, `3`, `4`, and `5`, the middle nodes are `0`, `1`, `1`, `2`, and `2`, respectively.

Example 1:



Input: head = [1,3,4,7,1,2,6]

4.5K | 77 | 61 Online

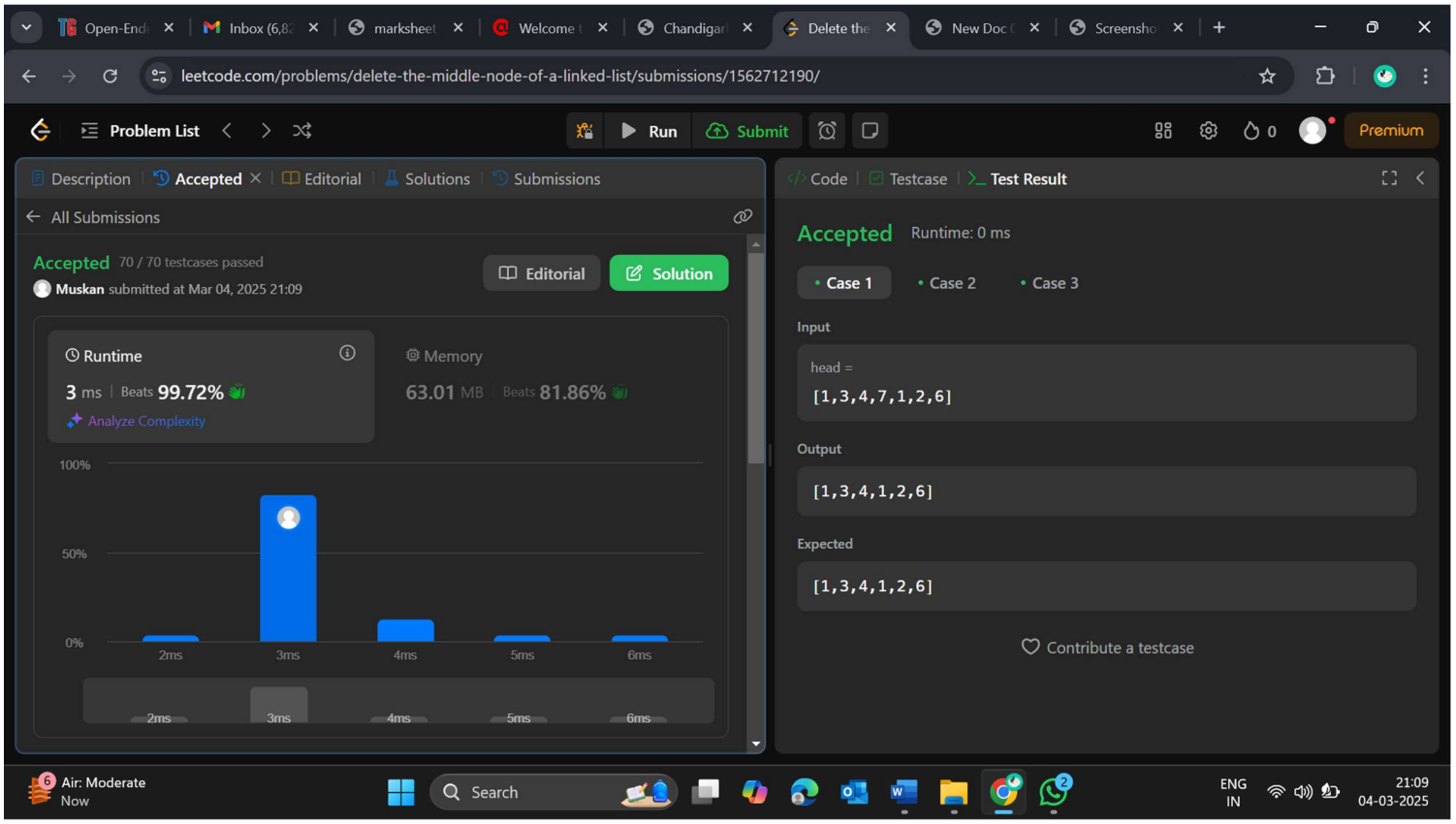
Code | Testcase | Test Result

Java | Auto

```
10  */
11  class Solution {
12      public ListNode deleteMiddle(ListNode head) {
13          int count = 0;
14          ListNode current = head;
15          while (current != null){
16              count++;
17              current = current.next;
18          }
19          current = head;
20          if(count == 1){
21              return current.next;
22          }
23          int n = (count / 2) - 1;
24          for(int i = 0; i < n; i++){
25              current = current.next;
26          }
27          current.next = current.next.next;
28          return head; //22BCS13343 MUSKAN
29      }
30  }
```

Saved | Ln 28, Col 41

Air: Moderate Now | Search | ENG IN | 21:09 04-03-2025


leetcode.com/problems/delete-the-middle-node-of-a-linked-list/submissions/1562712190/
Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 70 / 70 testcases passed
Muskan submitted at Mar 04, 2025 21:09
Editorial
Solution
Runtime
3 ms | Beats 99.72%
Analyze Complexity
Memory
63.01 MB | Beats 81.86%
100%
50%
0%
2ms
3ms
4ms
5ms
6ms
Code
Testcase
Test Result
Accepted Runtime: 0 ms
Case 1
Case 2
Case 3
Input
head =
[1,3,4,7,1,2,6]
Output
[1,3,4,1,2,6]
Expected
[1,3,4,1,2,6]
Contribute a testcase
Air: Moderate
Now
Search
ENG
IN
21:09
04-03-2025

Open-End | Inbox (6,8 | marksheet | Welcome t | Chandigar | Linked List | New Doc | Screensho

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

Problem List | Run | Submit | 0 | Premium

Description | Editorial | Solutions | Submissions

141. Linked List Cycle

Solved

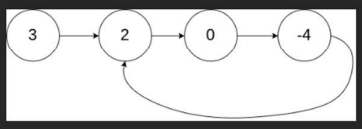
Easy | Topics | Companies

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:



16.2K | 360 | 240 Online

Code | Testcase | Test Result

C++ | Auto

```
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    bool hasCycle(ListNode* head) {
        ListNode* slow = head;
        ListNode* fast = head;
        while (fast != NULL && fast->next != NULL) {
            slow = slow->next;
            fast = fast->next->next;
            if (slow == fast) {
                return true;
            }
        }
        return false; //22BCS13343 MUSKAN
    }
};
```

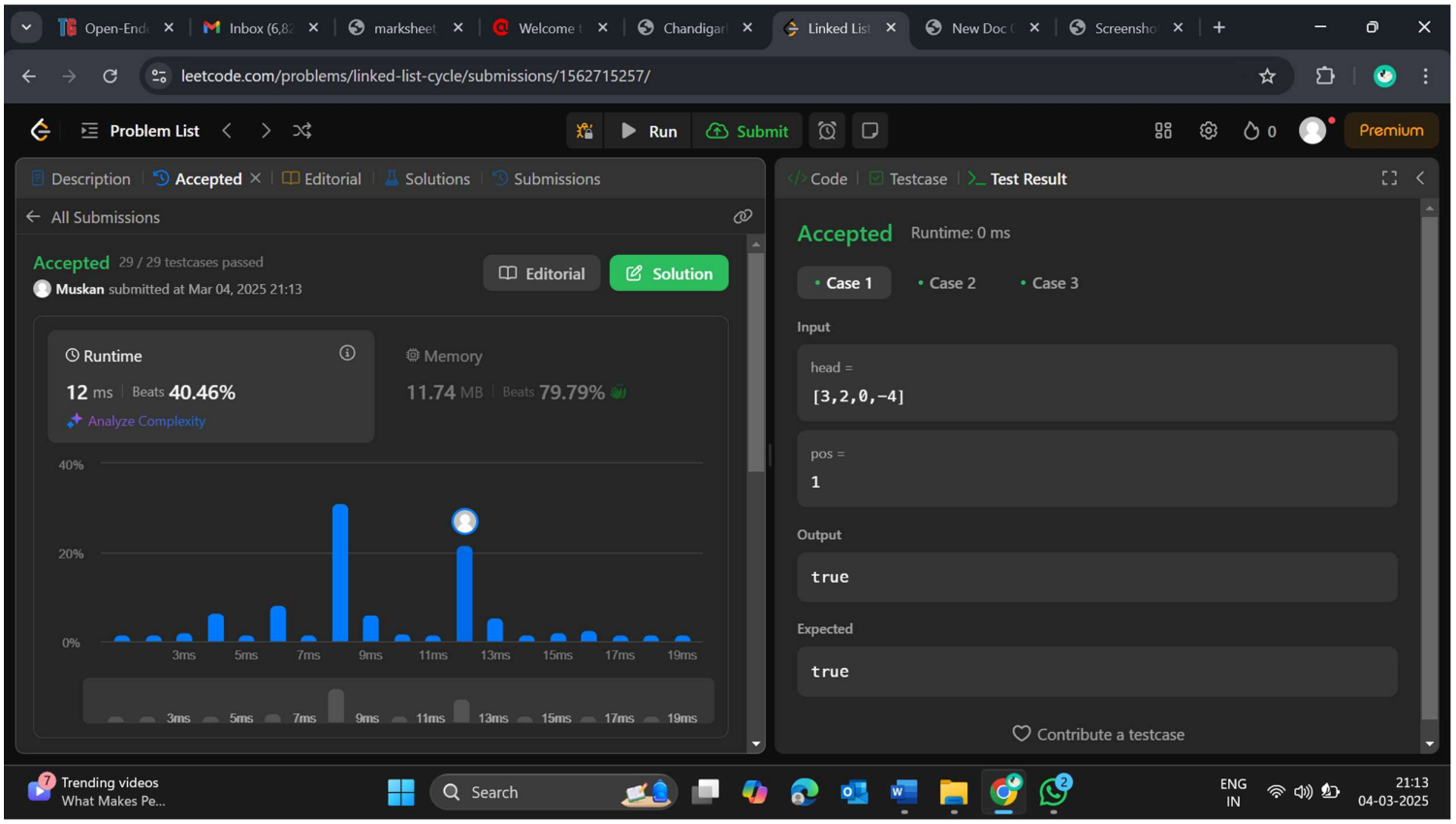
Saved | Ln 21, Col 42

Trending videos
What Makes Pe...

Search

ENG
IN

21:12
04-03-2025


leetcode.com/problems/linked-list-cycle/submissions/1562715257/
Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 29 / 29 testcases passed
Muskan submitted at Mar 04, 2025 21:13
Editorial
Solution
Runtime
12 ms | Beats 40.46%
Analyze Complexity
Memory
11.74 MB | Beats 79.79%
40%
20%
0%
3ms 5ms 7ms 9ms 11ms 13ms 15ms 17ms 19ms
Code
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
Trending videos
What Makes Pe...
Search
ENG
IN
21:13
04-03-2025

leetcode.com/problems/rotate-list/

Problem List | Run | Submit | Premium

Description | Editorial | Solutions | Submissions

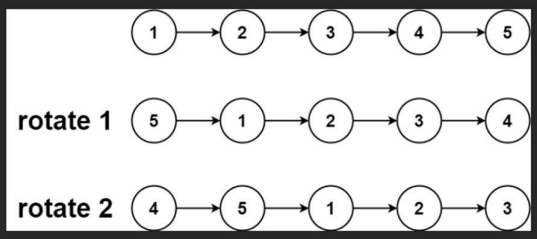
61. Rotate List

Solved

Medium | Topics | Companies

Given the `head` of a linked list, rotate the list to the right by `k` places.

Example 1:



Input: head = [1,2,3,4,5], k = 2

10.2K | 104 | 105 Online

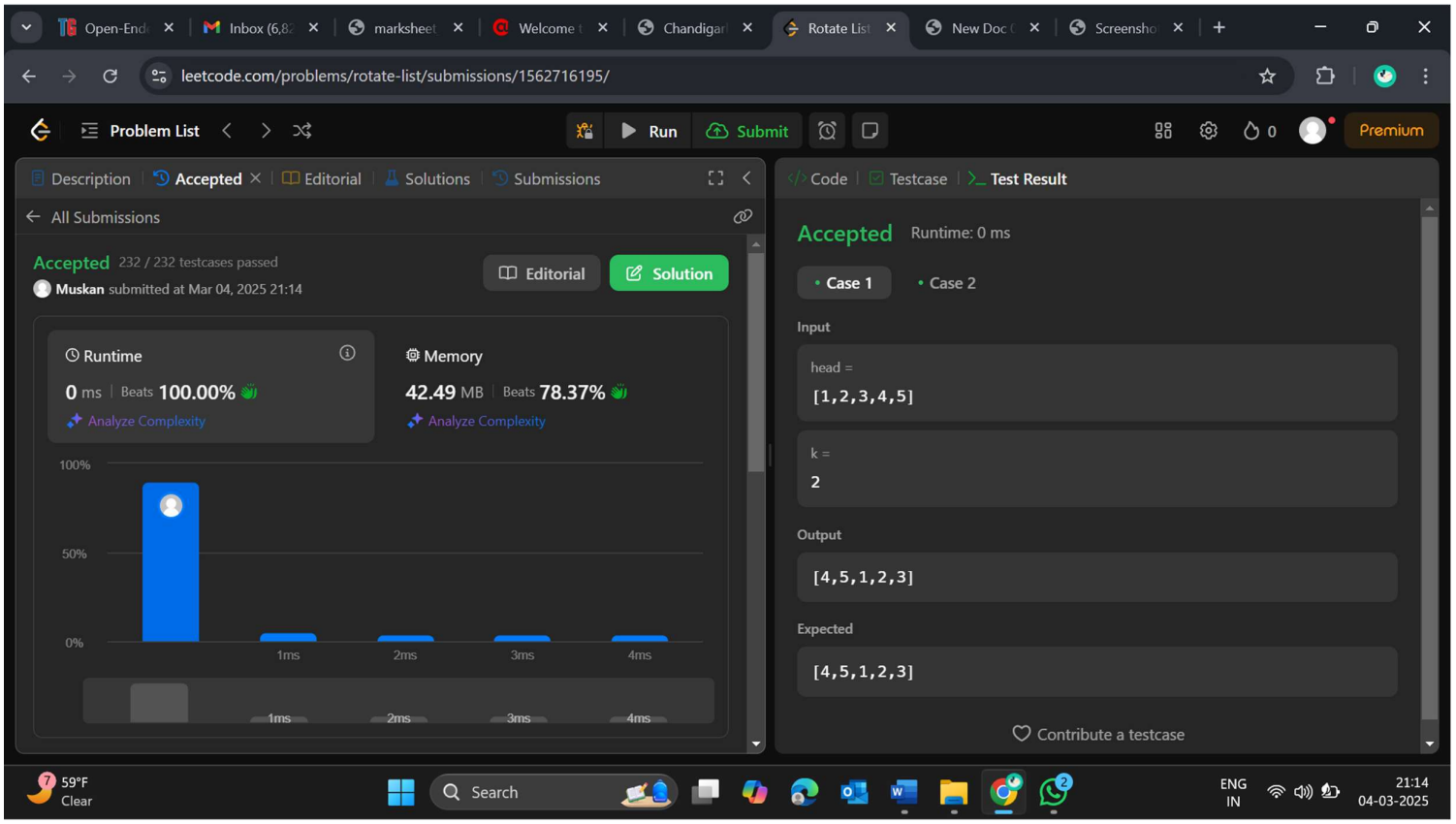
Code | Testcase | Test Result

Java | Auto

```java
class Solution {
    public ListNode rotateRight(ListNode head, int k) {
        if (head == null || head.next == null || k == 0) {
            return head;//22BCS13343 MUSKAN
        }
        ListNode temp = head;
        int length = 1;
        while (temp.next != null) {
            temp = temp.next;
            length++;
        }
        temp.next = head;
        k = k % length;
        int breakPoint = length - k;
        ListNode newTail = head;
        for (int i = 1; i < breakPoint; i++) {
            newTail = newTail.next;
        }
        head = newTail.next;
        newTail.next = null;
        return head;
    }
}
```

Saved | Ln 14, Col 44

59°F Clear | Search | ENG IN | 21:14 04-03-2025


leetcode.com/problems/rotate-list/submissions/1562716195/
Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 232 / 232 testcases passed
Muskan submitted at Mar 04, 2025 21:14
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
42.49 MB | Beats 78.37%
Analyze Complexity
100%
50%
0%
1ms
2ms
3ms
4ms
Code
Testcase
Test Result
Accepted Runtime: 0 ms
Case 1
Case 2
Input
head =
[1,2,3,4,5]
k =
2
Output
[4,5,1,2,3]
Expected
[4,5,1,2,3]
Contribute a testcase
59°F
Clear
Search
ENG
IN
21:14
04-03-2025

Problem List | Run | Submit | Premium

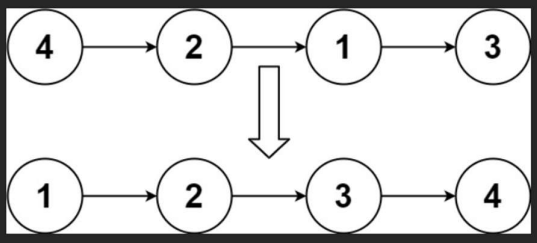
Description | Editorial | Solutions | Submissions

148. Sort List

Medium | Topics | Companies

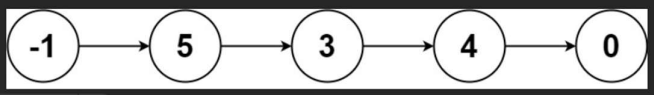
Given the `head` of a linked list, return *the list after sorting it in **ascending order***.

Example 1:




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

Example 2:



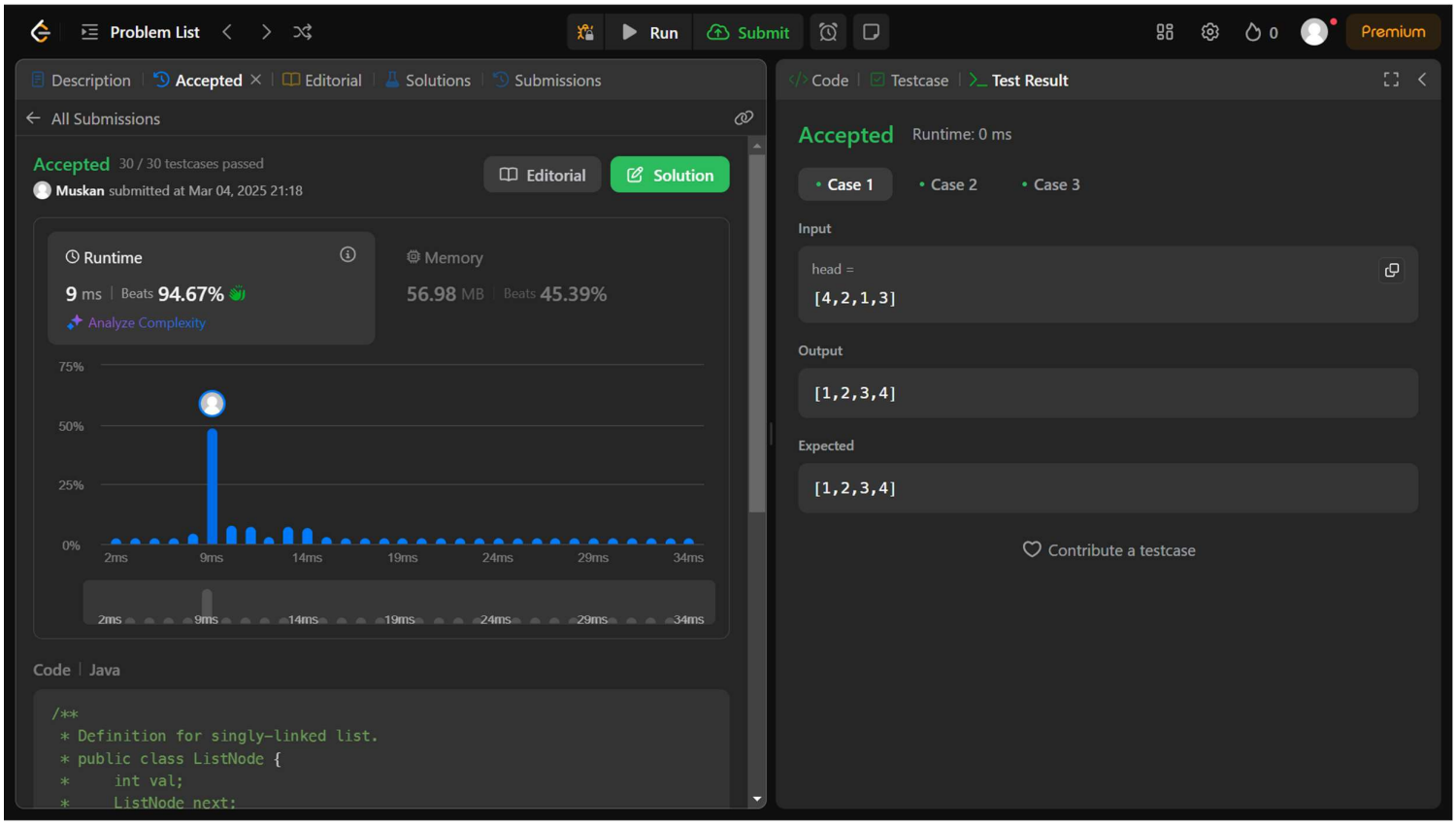
12.2K | 111 | 107 Online

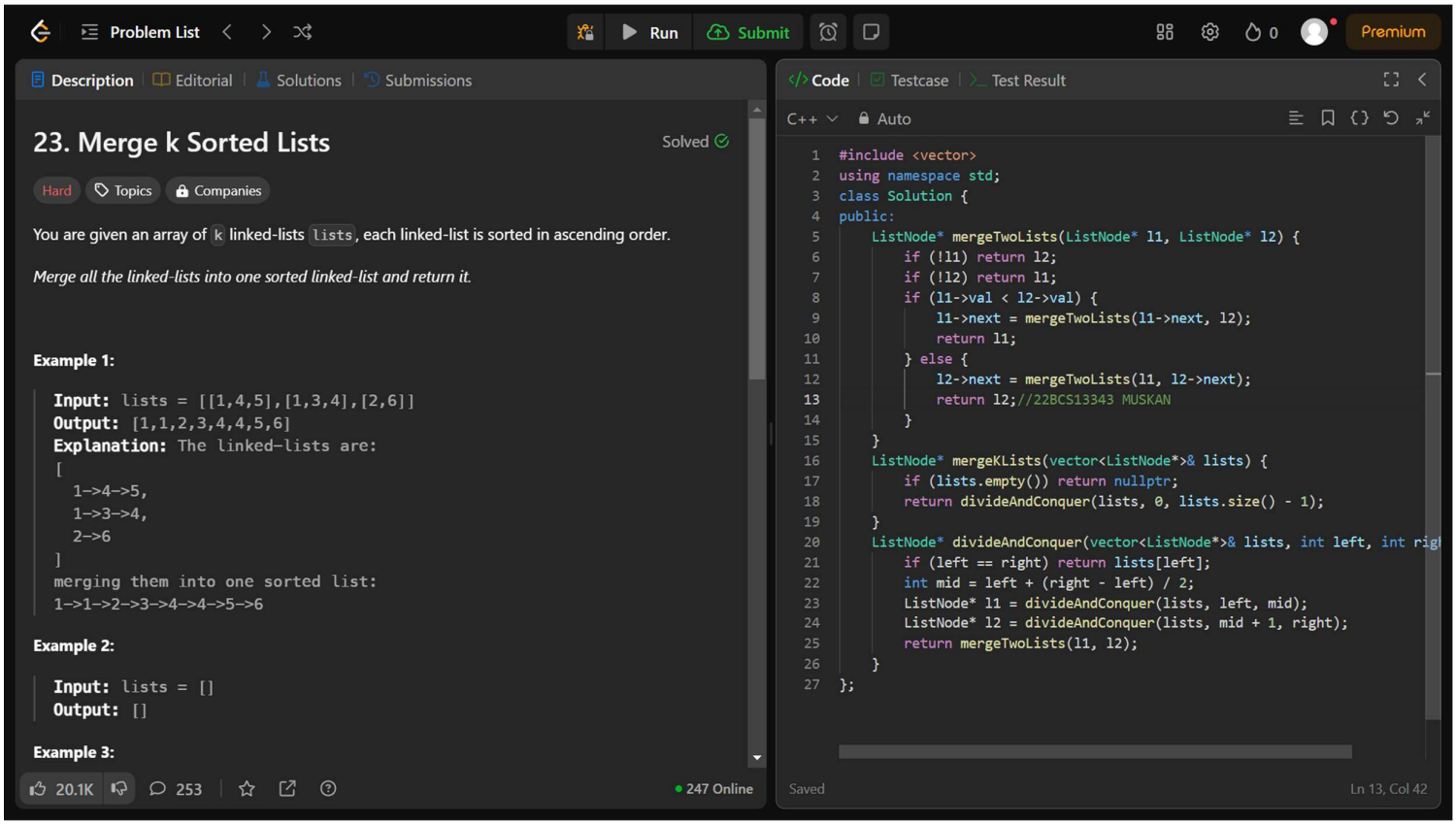
Code | Testcase | Test Result

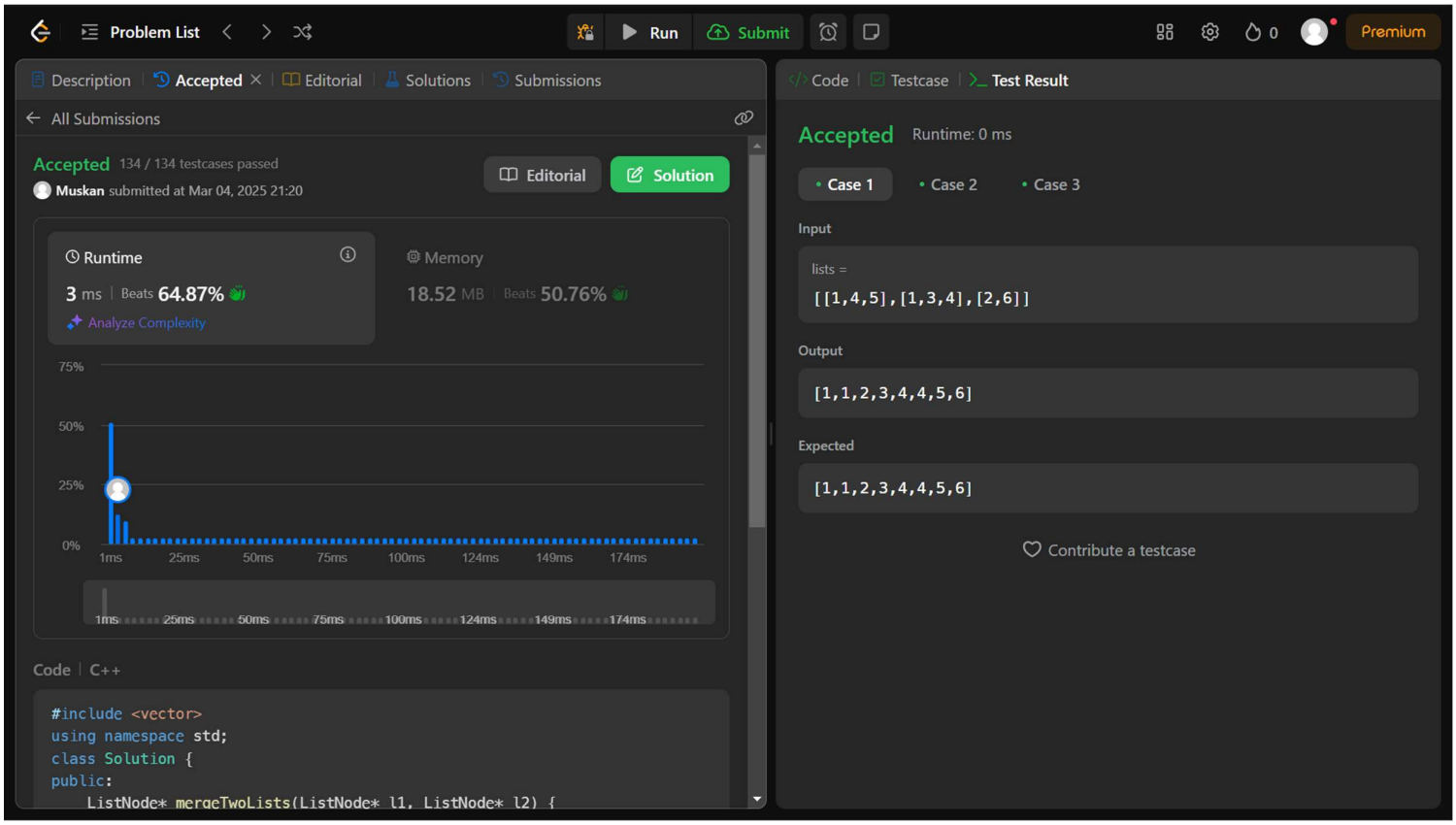
Java | Auto

```java
class Solution {
    public ListNode sortList(ListNode head) {
        if (head == null || head.next == null) return head;
        ListNode slow = head, fast = head.next;
        while (fast != null && fast.next != null) {
            slow = slow.next;
            fast = fast.next.next;
        }
        ListNode mid = slow.next;
        slow.next = null;
        ListNode left = sortList(head);//22BCS13343 MUSKAN
        ListNode right = sortList(mid);
        return merge(left, right);
    }
    private ListNode merge(ListNode l1, ListNode l2) {
        ListNode dummy = new ListNode(0);
        ListNode tail = dummy;
        while (l1 != null && l2 != null) {
            if (l1.val < l2.val) {
                tail.next = l1;
                l1 = l1.next;
            } else {
                tail.next = l2;
                l2 = l2.next;
            }
            tail = tail.next;
        }
        tail.next = (l1 != null) ? l1 : l2;
        return dummy.next;
    }
}
```

Saving... | Ln 21, Col 59


Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 30 / 30 testcases passed
Muskan submitted at Mar 04, 2025 21:18
Editorial
Solution
Runtime
9 ms | Beats 94.67%
Analyze Complexity
Memory
56.98 MB | Beats 45.39%
75%
50%
25%
0%
2ms
9ms
14ms
19ms
24ms
29ms
34ms
Code | Java
/**
* Definition for singly-linked list.
* public class ListNode {
* int val;
* ListNode next;
Code
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


Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
Code
Testcase
Test Result
C++
Auto
23. Merge k Sorted Lists
Solved
Hard
Topics
Companies
You are given an array of k linked-lists lists, each linked-list is sorted in ascending order.
Merge all the linked-lists into one sorted linked-list and return it.
Example 1:
Input: lists = [[1,4,5],[1,3,4],[2,6]]
Output: [1,1,2,3,4,4,5,6]
Explanation: The linked-lists are:
[
1->4->5,
1->3->4,
2->6
]
merging them into one sorted list:
1->1->2->3->4->4->5->6
Example 2:
Input: lists = []
Output: []
Example 3:
20.1K
253
247 Online
#include <vector>
using namespace std;
class Solution {
public:
ListNode* mergeTwoLists(ListNode* l1, ListNode* l2) {
if (!l1) return l2;
if (!l2) return l1;
if (l1->val < l2->val) {
l1->next = mergeTwoLists(l1->next, l2);
return l1;
} else {
l2->next = mergeTwoLists(l1, l2->next);
return l2;//22BCS13343 MUSKAN
}
}
ListNode* mergeKLists(vector<ListNode*>& lists) {
if (lists.empty()) return nullptr;
return divideAndConquer(lists, 0, lists.size() - 1);
}
ListNode* divideAndConquer(vector<ListNode*>& lists, int left, int rig
if (left == right) return lists[left];
int mid = left + (right - left) / 2;
ListNode* l1 = divideAndConquer(lists, left, mid);
ListNode* l2 = divideAndConquer(lists, mid + 1, right);
return mergeTwoLists(l1, l2);
}
};
Saved
Ln 13, Col 42


Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 134 / 134 testcases passed
Muskan submitted at Mar 04, 2025 21:20
Editorial
Solution
Runtime
3 ms | Beats 64.87%
Analyze Complexity
Memory
18.52 MB | Beats 50.76%
75%
50%
25%
0%
1ms 25ms 50ms 75ms 100ms 124ms 149ms 174ms
Code | C++
#include <vector>
using namespace std;
class Solution {
public:
ListNode* mergeTwoLists(ListNode* l1, ListNode* l2) {
Code
Testcase
Test Result
Accepted Runtime: 0 ms
Case 1
Case 2
Case 3
Input
lists =
[[1,4,5],[1,3,4],[2,6]]
Output
[1,1,2,3,4,4,5,6]
Expected
[1,1,2,3,4,4,5,6]
Contribute a testcase