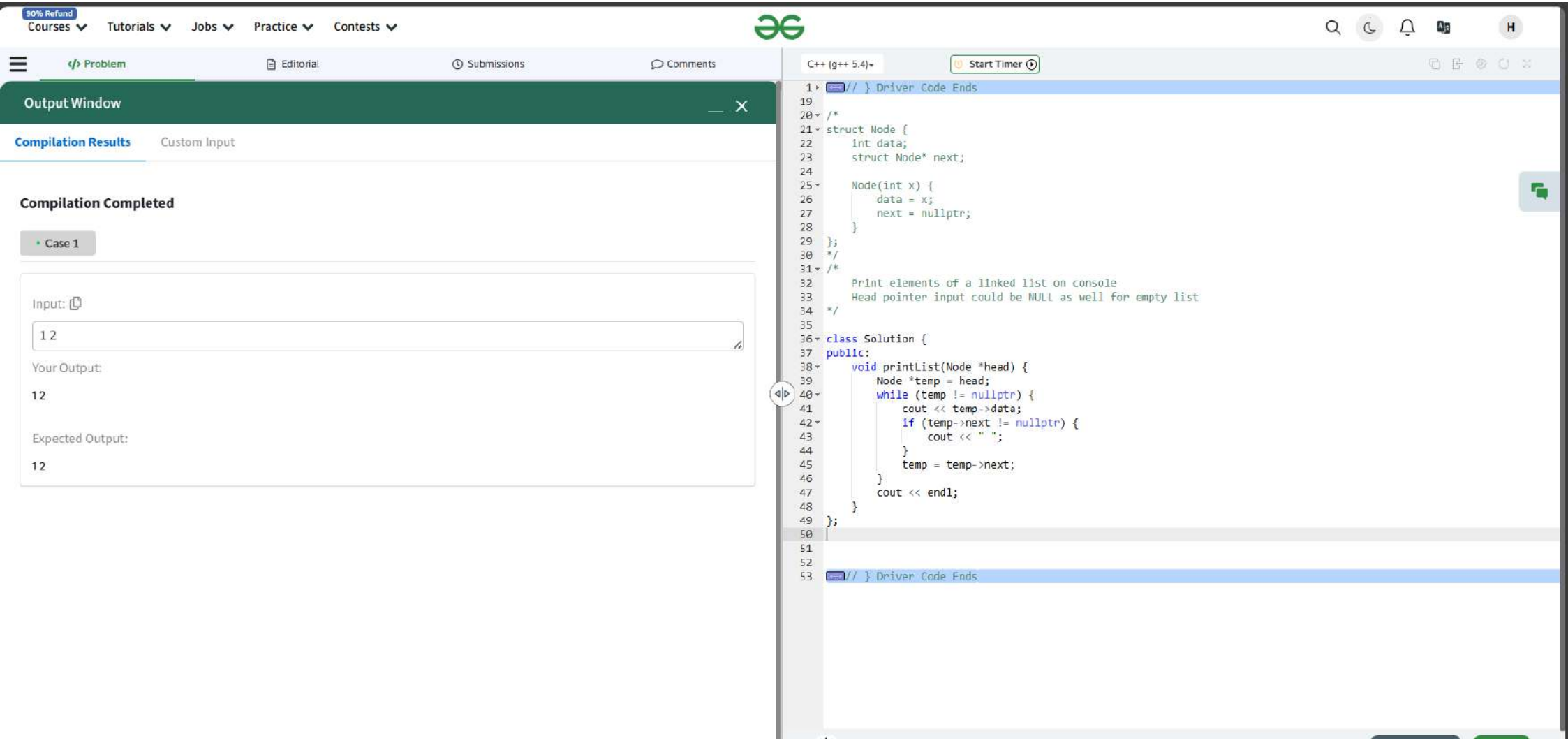

50% Refund
Courses
Tutorials
Jobs
Practice
Contests
Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Compilation Completed
Case 1
Input:
1 2
Your Output:
1 2
Expected Output:
1 2
C++ (g++ 5.4)
Start Timer
// } Driver Code Ends
/*
struct Node {
int data;
struct Node* 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:
void printList(Node *head) {
Node *temp = head;
while (temp != nullptr) {
cout << temp->data;
if (temp->next != nullptr) {
cout << " ";
}
temp = temp->next;
}
cout << endl;
}
};
// } Driver Code Ends

Remove Duplicates from Sorted

leetcode.com/problems/remove-duplicates-from-sorted-list/submissions/1565191748/

Gmail | YouTube | Maps | All Bookmarks

Problem List | Run | Submit | Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 168 / 168 testcases passed

HrishabhGupta7292 submitted at Mar 06, 2025 23:50

Editorial | Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

16.27 MB | Beats 35.10%

Code | C++

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

Code

C++ | Auto

```
12  public:
13      ListNode* deleteDuplicates(ListNode* head) {
14          ListNode* current = head;
15
16          while (current != nullptr && current->next != nullptr) {
17              if (current->val == current->next->val) {
18
19                  ListNode* temp = current->next;
20                  current->next = current->next->next;
21                  delete temp;
22              } else {
23
24                  current = current->next;
25              }
26          }
27
28          return head;
29      }
30  };
31
```

Saved | Ln 31, Col 1

Testcase | Test Result

Case 1 | Case 2 | +

head =

Source

23:50
06-03-2025

Reverse Linked List - LeetCode

leetcode.com/problems/reverse-linked-list/submissions/1565192675/

Gmail | YouTube | Maps | All Bookmarks

Problem List | Run | Submit | Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 28 / 28 testcases passed

HrishabhGupta7292 submitted at Mar 06, 2025 23:50

Editorial | Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

13.46 MB | Beats 39.75%

Code | C++

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

Code

C++ | Auto

```
 */
class Solution {
public:
    ListNode* reverseList(ListNode* head) {
        ListNode* prev = nullptr;
        ListNode* current = head;
        ListNode* next = nullptr;

        while (current != nullptr) {
            next = current->next;
            current->next = prev;
            prev = current;
            current = next;
        }

        return prev;
    }
};
```

Saved | Ln 28, Col 1

Testcase | Test Result

Case 1 | Case 2 | Case 3

head =

Source

23:51
06-03-2025

Delete the Middle Node of a Li

leetcode.com/problems/delete-the-middle-node-of-a-linked-list/submissions/1565193242/

Gmail YouTube Maps All Bookmarks

Problem List Run Submit 0 Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 70 / 70 testcases passed

HrishabhGupta7292 submitted at Mar 06, 2025 23:51

Editorial Solution

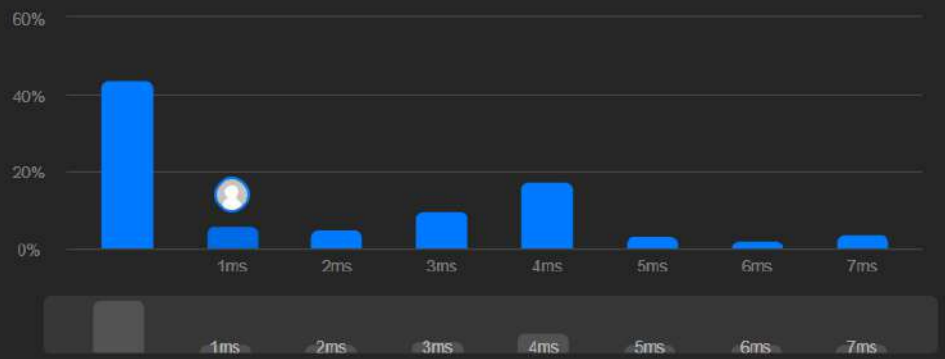
Runtime

1 ms | Beats 56.17%

Analyze Complexity

Memory

312.18 MB | Beats 17.89%



Code | C++

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

Code

C++ Auto

```
class Solution {
public:
    ListNode* deleteMiddle(ListNode* head) {
        if (!head || !head->next) return nullptr;

        ListNode* slow = head;
        ListNode* fast = head;
        ListNode* prev = nullptr;

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

        prev->next = slow->next;
        delete slow;

        return head;
    }
};
```

Saved Ln 32, Col 1

Testcase | Test Result

Case 1 Case 2 Case 3

head =

Source

23:51
06-03-2025

Merge Two Sorted Lists - LeetC

leetcode.com/problems/merge-two-sorted-lists/submissions/1565194375/

Gmail YouTube Maps All Bookmarks

Problem List Run Submit 0 Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 208 / 208 testcases passed

HrishabhGupta7292 submitted at Mar 06, 2025 23:52

Editorial Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

19.40 MB | Beats 86.51%

Code | C++

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

Code

C++ Auto

```
13      ListNode* mergeTwoLists(ListNode* list1, ListNode* list2) {
14          if (!list1) return list2;
15          if (!list2) return list1;
16
17          ListNode* dummy = new ListNode(-1);
18          ListNode* current = dummy;
19
20          while (list1 && list2) {
21              if (list1->val <= list2->val) {
22                  current->next = list1;
23                  list1 = list1->next;
24              } else {
25                  current->next = list2;
26                  list2 = list2->next;
27              }
28              current = current->next;
29          }
30
31          if (list1) current->next = list1;
32          if (list2) current->next = list2;
33
34          ListNode* mergedHead = dummy->next;
35          delete dummy;
36          return mergedHead;
```

Saved Ln 39, Col 1

Testcase | Test Result

Source

23:52
06-03-2025

Linked List Cycle - LeetCode

leetcode.com/problems/linked-list-cycle/submissions/1565195658/

Gmail | YouTube | Maps | All Bookmarks

Problem List | Run | Submit | Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 29 / 29 testcases passed

HrishabhGupta7292 submitted at Mar 06, 2025 23:53

Editorial | Solution

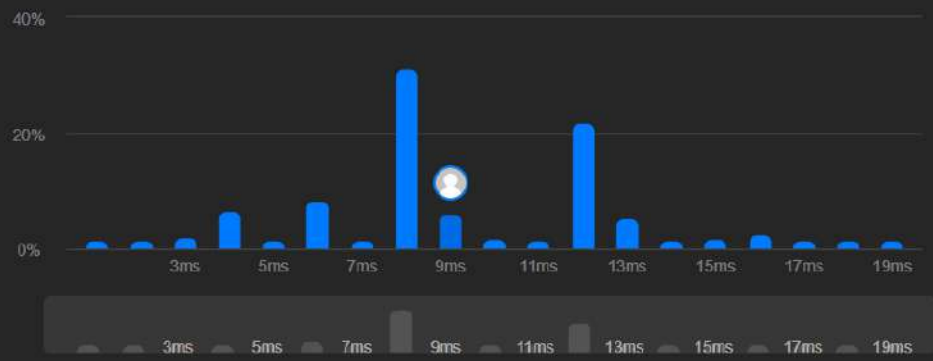
Runtime

9 ms | Beats 49.74%

Analyze Complexity

Memory

11.97 MB | Beats 24.19%



Code | C++

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

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 || !head->next) return false;

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

        while (fast && fast->next) {
            slow = slow->next;
            fast = fast->next->next;
            if (slow == fast) return true;
        }

        return false;
    }
};
```

Saved

Ln 26, Col 1

Testcase | Test Result

Source

23:53
06-03-2025

Rotate List - LeetCode

leetcode.com/problems/rotate-list/submissions/1565196230/

Gmail YouTube Maps All Bookmarks

Problem List Run Submit 0 Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 232 / 232 testcases passed

HrishabhGupta7292 submitted at Mar 06, 2025 23:54

Editorial Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

16.46 MB | Beats 31.91%

Code | C++

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

Code

C++ Auto

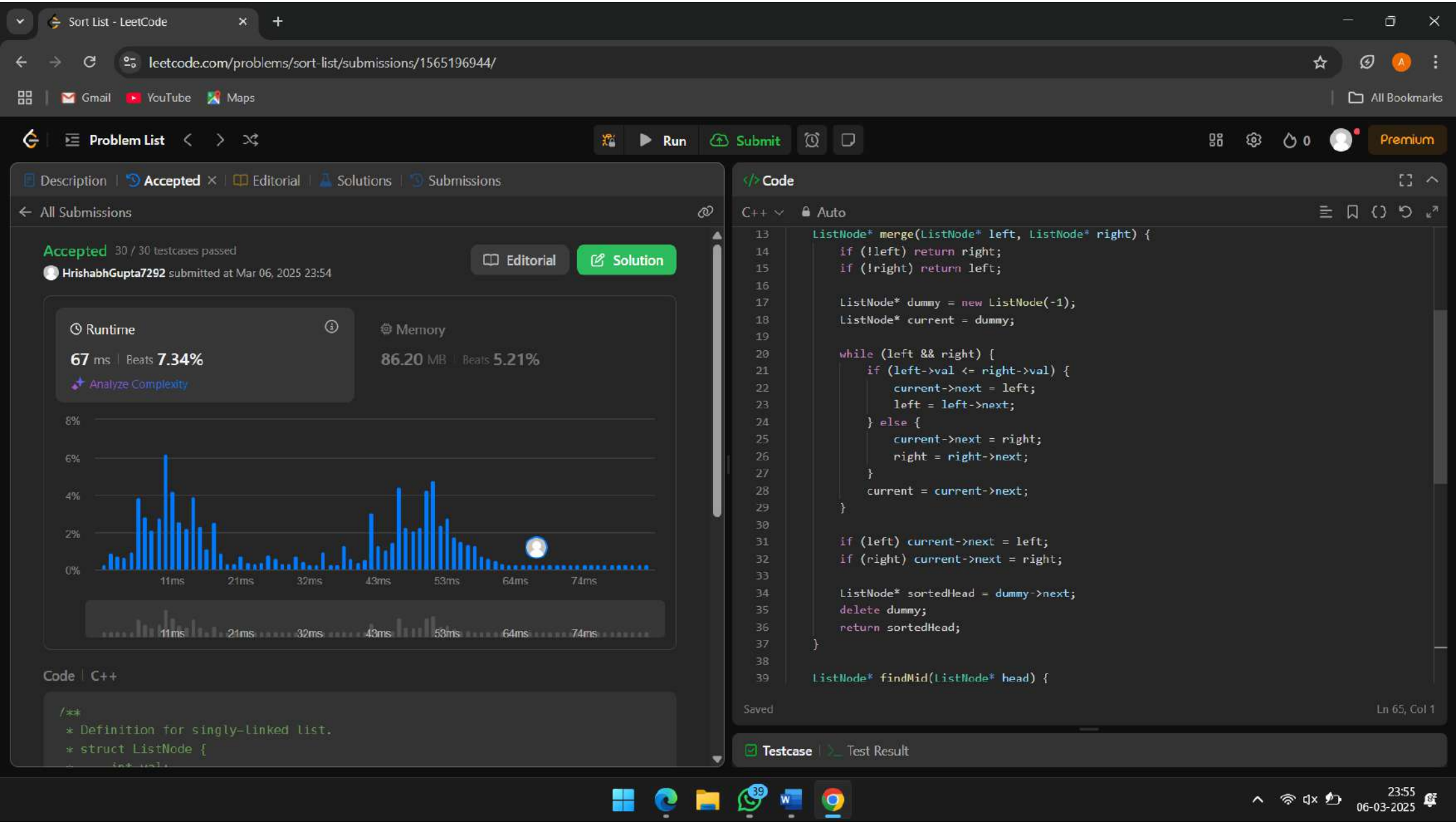
```
class Solution {
public:
    ListNode* rotateRight(ListNode* head, int k) {
        if (!head || !head->next || k == 0) return head;

        ListNode* temp = head;
        int length = 1;

        while (temp->next) {
            temp = temp->next;
            length++;
        }

        temp->next = head;
        k = k % length;
        int stepsToNewHead = length - k;

        while (stepsToNewHead--) {
            temp = temp->next;
        }

        head = temp->next;
        temp->next = nullptr;

        return head;
```

Saved Ln 38, Col 1

Testcase | Test Result

Source

23:54
06-03-2025


Sort List - LeetCode
leetcode.com/problems/sort-list/submissions/1565196944/
Gmail
YouTube
Maps
All Bookmarks
Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 30 / 30 testcases passed
HrishabhGupta7292 submitted at Mar 06, 2025 23:54
Editorial
Solution
Runtime
67 ms | Beats 7.34%
Analyze Complexity
Memory
86.20 MB | Beats 5.21%
8%
6%
4%
2%
0%
11ms
21ms
32ms
43ms
53ms
64ms
74ms
Code | C++
/**
* Definition for singly-linked list.
* struct ListNode {
Code
C++
Auto
ListNode* merge(ListNode* left, ListNode* right) {
if (!left) return right;
if (!right) return left;
ListNode* dummy = new ListNode(-1);
ListNode* current = dummy;
while (left && right) {
if (left->val <= right->val) {
current->next = left;
left = left->next;
} else {
current->next = right;
right = right->next;
}
current = current->next;
}
if (left) current->next = left;
if (right) current->next = right;
ListNode* sortedHead = dummy->next;
delete dummy;
return sortedHead;
}
ListNode* findMid(ListNode* head) {
Saved
Ln 65, Col 1
Testcase | Test Result
23:55
06-03-2025

Merge k Sorted Lists - LeetCod

leetcode.com/problems/merge-k-sorted-lists/submissions/1565197881/

Gmail YouTube Maps All Bookmarks

Problem List Run Submit 0 Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 134 / 134 testcases passed

HrishabhGupta7292 submitted at Mar 06, 2025 23:55

Editorial Solution

Runtime

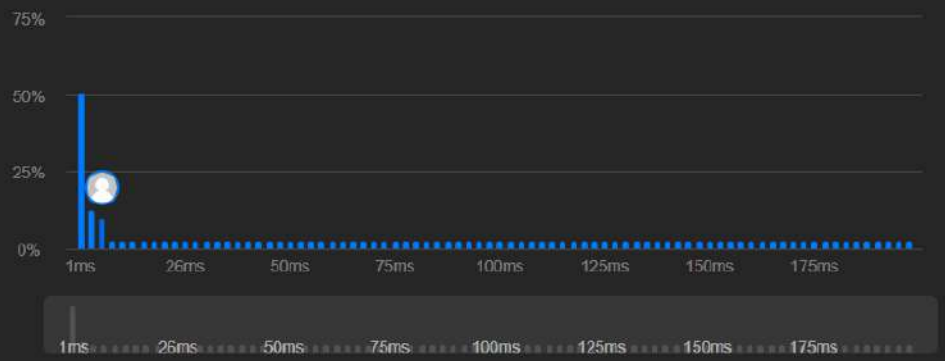
7 ms | Beats 34.03%

Analyze Complexity

Memory

18.28 MB | Beats 89.49%

Analyze Complexity



Code | C++

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

Code

C++ Auto

```
        return a->val > b->val;
    }
};

ListNode* mergeKLists(vector<ListNode*>& lists) {
    priority_queue<ListNode*, vector<ListNode*>, Compare> minHeap;

    for (auto list : lists) {
        if (list) minHeap.push(list);
    }

    ListNode* dummy = new ListNode(-1);
    ListNode* current = dummy;

    while (!minHeap.empty()) {
        ListNode* smallest = minHeap.top();
        minHeap.pop();

        current->next = smallest;
        current = current->next;

        if (smallest->next) minHeap.push(smallest->next);
    }

    ListNode* mergedHead = dummy->next;
    delete dummy;
    return mergedHead;
```

Saved Ln 44, Col 1

Testcase | Test Result

23:55
06-03-2025