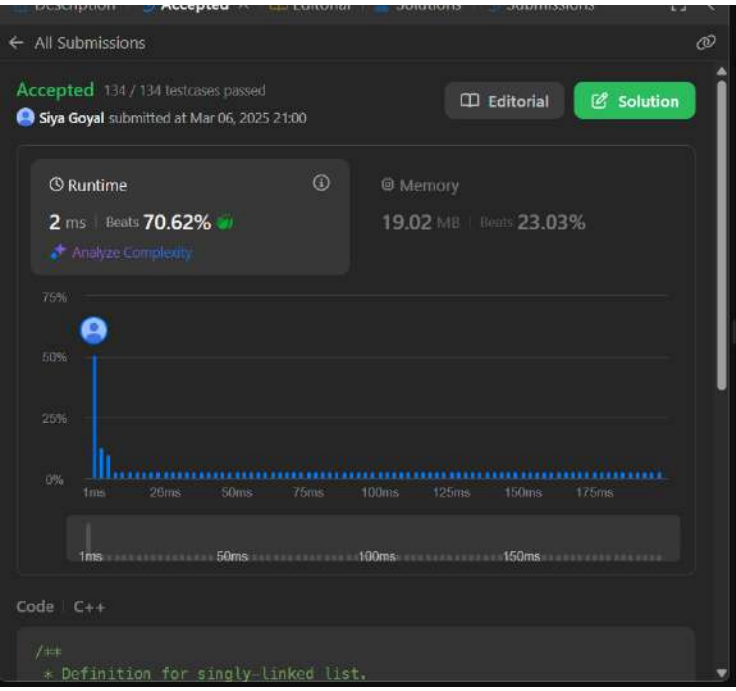
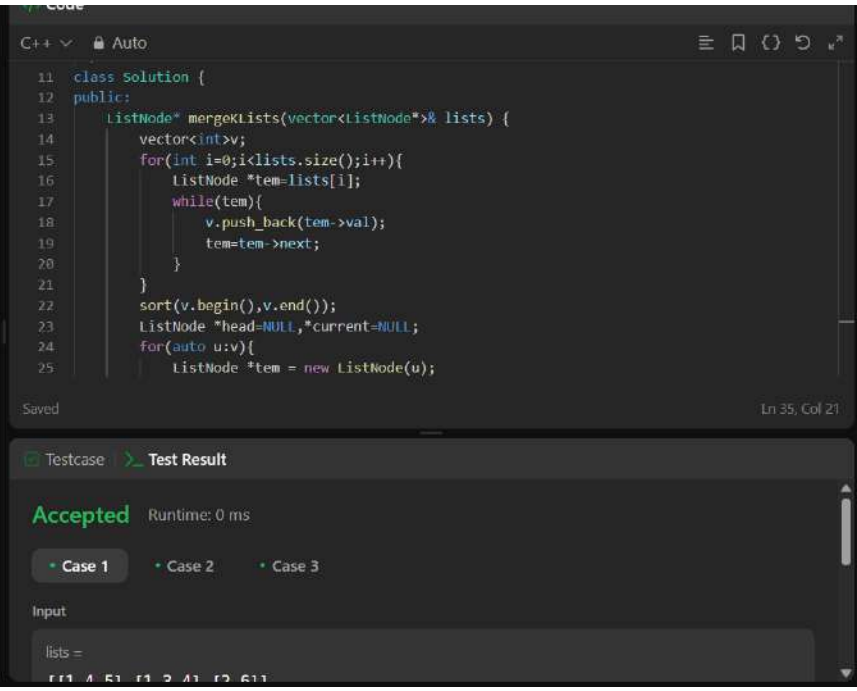
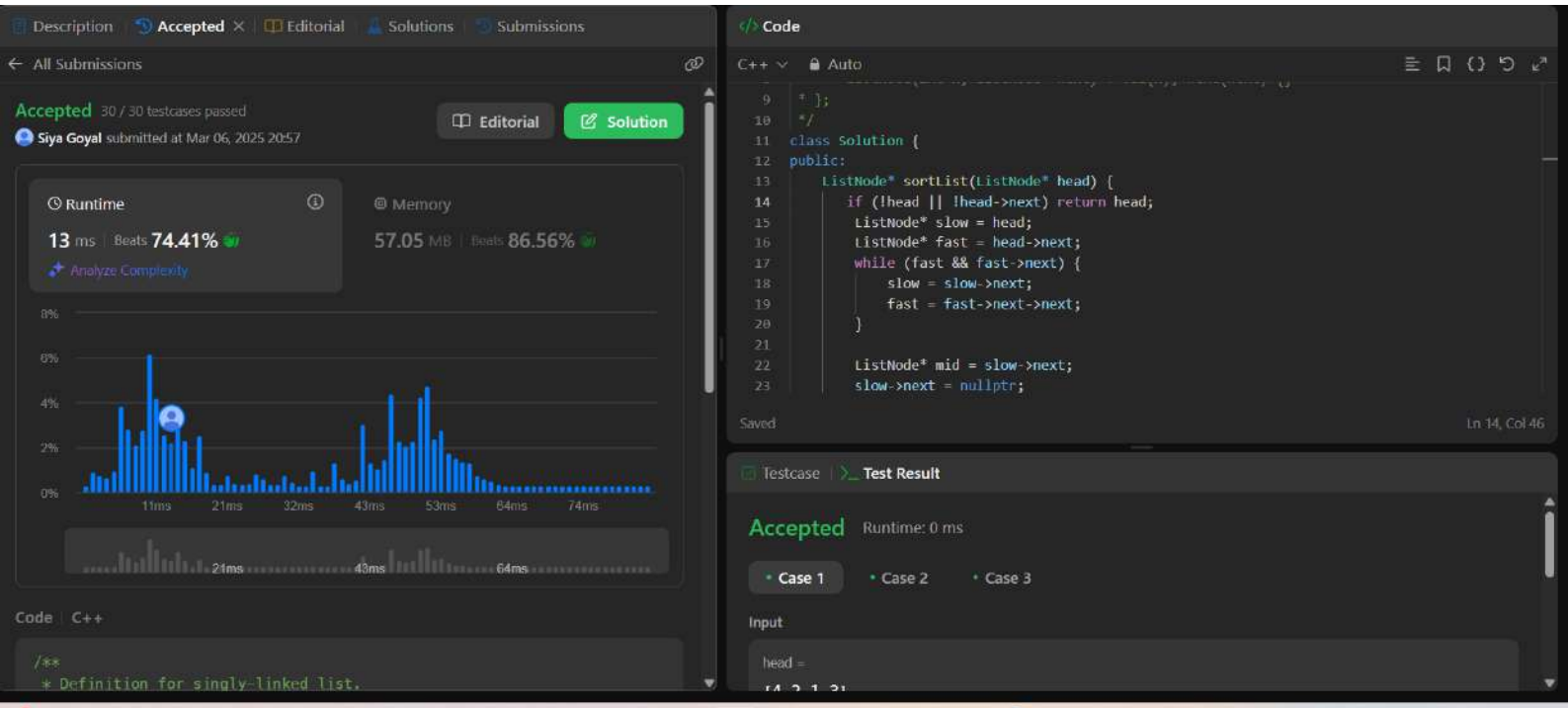

All Submissions
Accepted 134 / 134 testcases passed
Siya Goyal submitted at Mar 06, 2025 21:00
Editorial
Solution
Runtime
2 ms | Beats 70.62%
Analyze Complexity
Memory
19.02 MB | Beats 23.03%
75%
50%
25%
0%
1ms 25ms 50ms 75ms 100ms 125ms 150ms 175ms
1ms 50ms 100ms 150ms
Code | C++
/**
* Definition for singly-linked list.


C++ Auto
class Solution {
public:
ListNode* mergeKLists(vector<ListNode*>& lists) {
vector<int>v;
for(int i=0;i<lists.size();i++){
ListNode *tem=lists[i];
while(tem){
v.push_back(tem->val);
tem=tem->next;
}
}
sort(v.begin(),v.end());
ListNode *head=NULL,*current=NULL;
for(auto u:v){
ListNode *tem = new ListNode(u);
Saved
Ln 35, Col 21
Testcase | Test Result
Accepted Runtime: 0 ms
Case 1
Case 2
Case 3
Input
lists =


Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 30 / 30 testcases passed
Siya Goyal submitted at Mar 06, 2025 20:57
Editorial
Solution
Runtime
13 ms | Beats 74.41%
Analyze Complexity
Memory
57.05 MB | Beats 86.56%
8%
6%
4%
2%
0%
11ms
21ms
32ms
43ms
53ms
64ms
74ms
21ms
43ms
64ms
Code | C++
/**
* Definition for singly-linked list.
Code
C++
Auto
* };
*/
class Solution {
public:
ListNode* sortList(ListNode* head) {
if (!head || !head->next) return head;
ListNode* slow = head;
ListNode* fast = head->next;
while (fast && fast->next) {
slow = slow->next;
fast = fast->next->next;
}
ListNode* mid = slow->next;
slow->next = nullptr;
Saved
Ln 14, Col 46
Testcase | Test Result
Accepted Runtime: 0 ms
Case 1
Case 2
Case 3
Input
head =

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 232 / 232 testcases passed

Siya Goyal submitted at Mar 06, 2025 20:56

Editorial | Solution

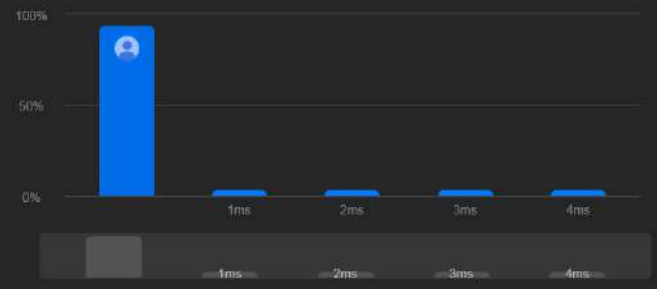
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

16.21 MB | Beats 93.90%



Code | C++

```
/**
 * Definition for singly-linked list.
```

Code

C++ | Auto

```
 * };
 */
class Solution {
public:
    ListNode* rotateRight(ListNode* head, int k) {
        if (!head || !head->next || k == 0) return head;
        ListNode* current = head;
        int length = 1;
        while (current->next)
        {
            length++;
            current = current->next;
        }

        k %= length;
```

Saved

Ln 21, Col 10

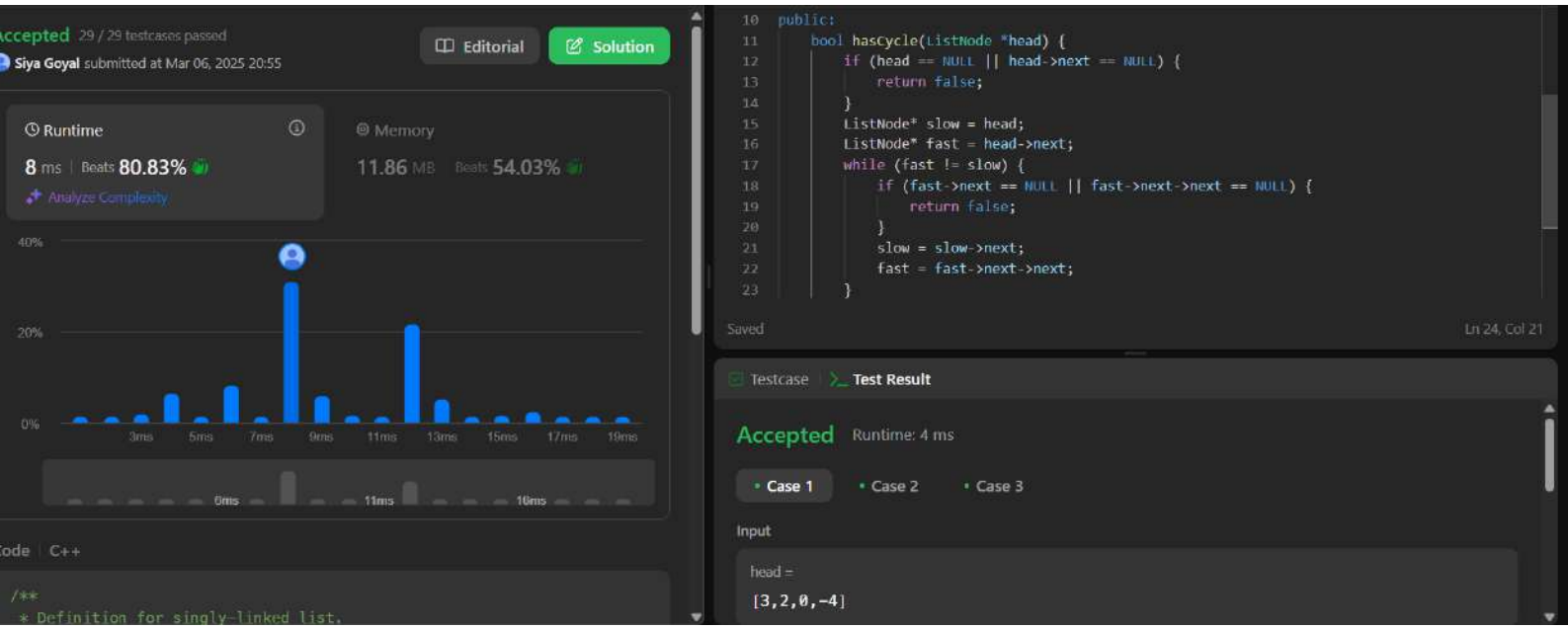
Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2

Input

head =




```cpp
public:
    bool hasCycle(ListNode *head) {
        if (head == NULL || head->next == NULL) {
            return false;
        }
        ListNode* slow = head;
        ListNode* fast = head->next;
        while (fast != slow) {
            if (fast->next == NULL || fast->next->next == NULL) {
                return false;
            }
            slow = slow->next;
            fast = fast->next->next;
        }
```

Saved Ln 24, Col 21

Testcase | Test Result

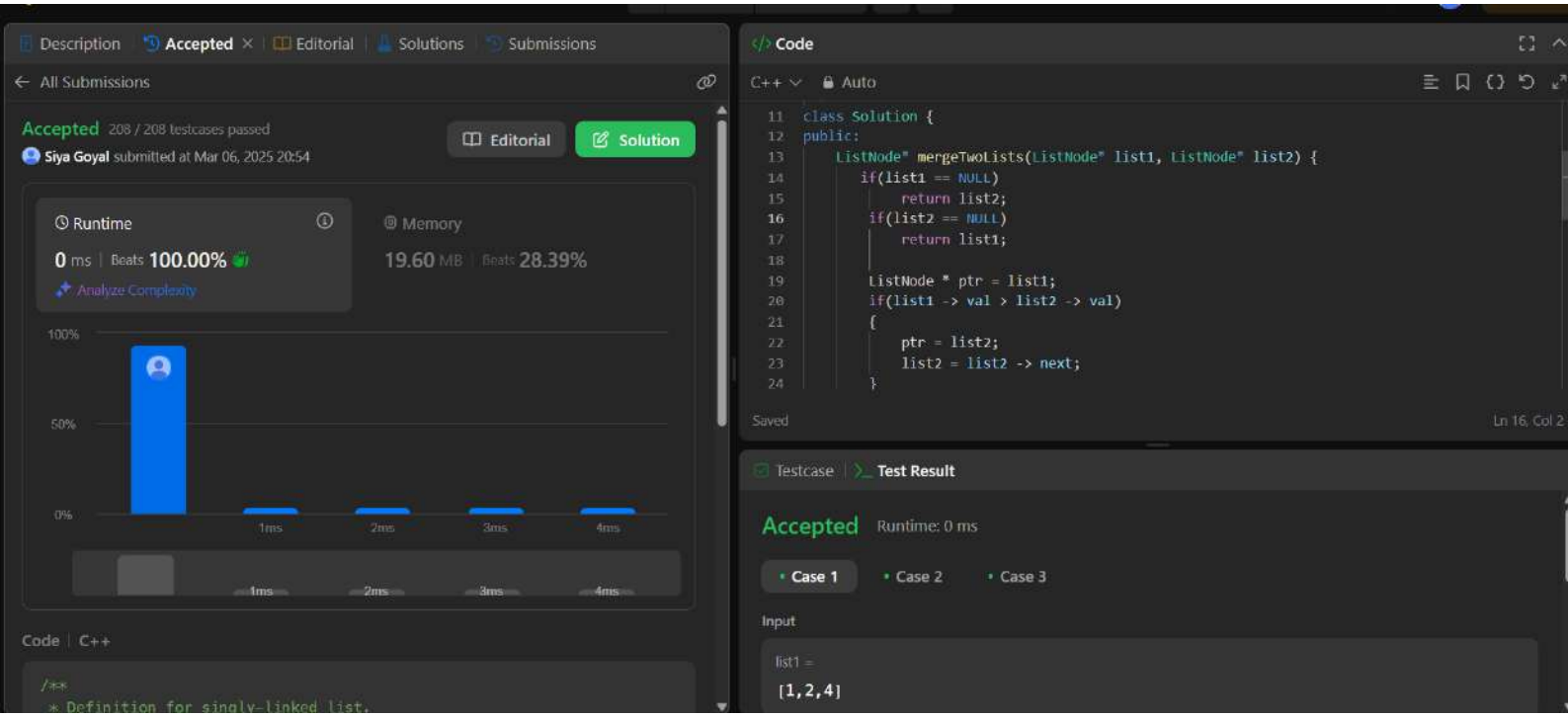
Accepted Runtime: 4 ms

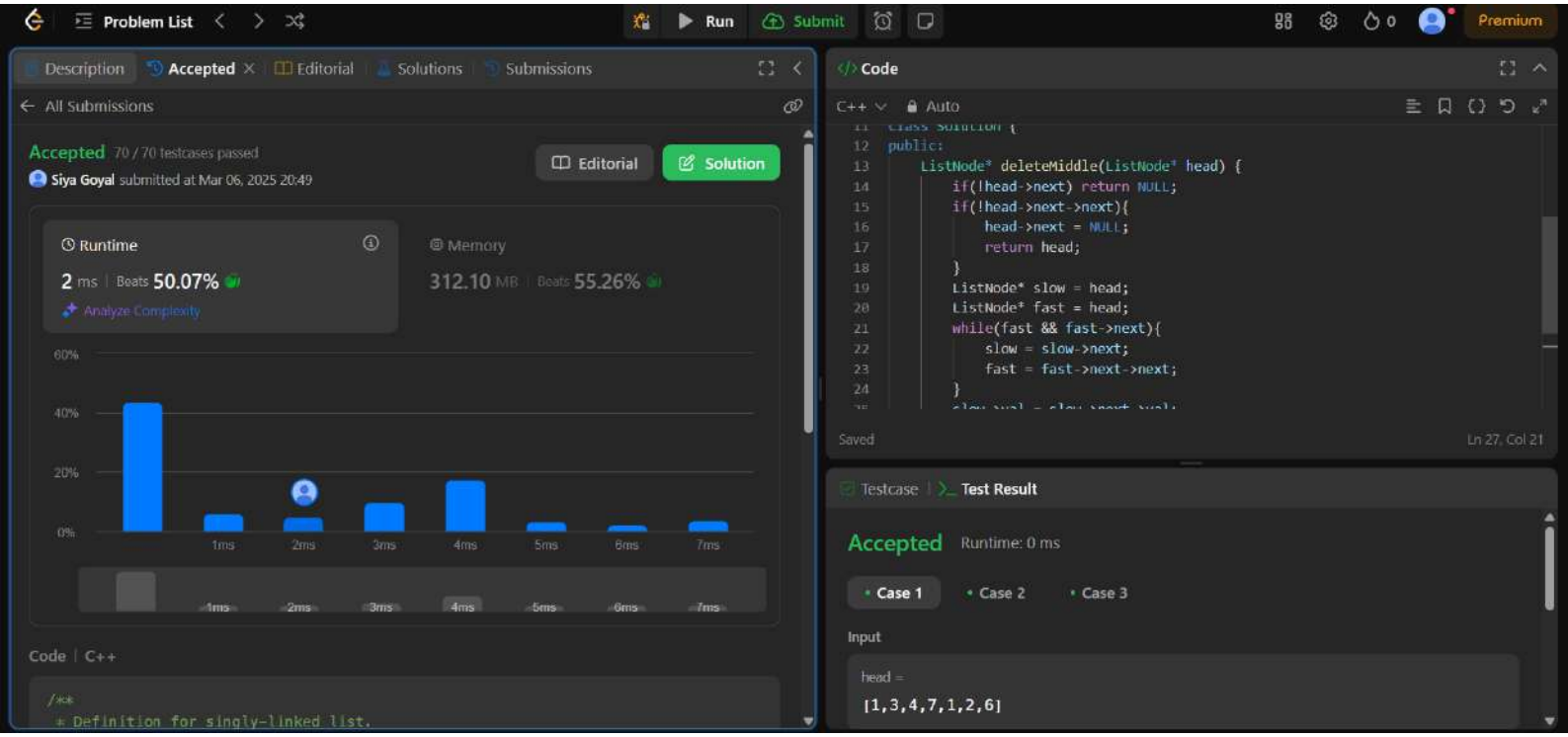
• Case 1 • Case 2 • Case 3

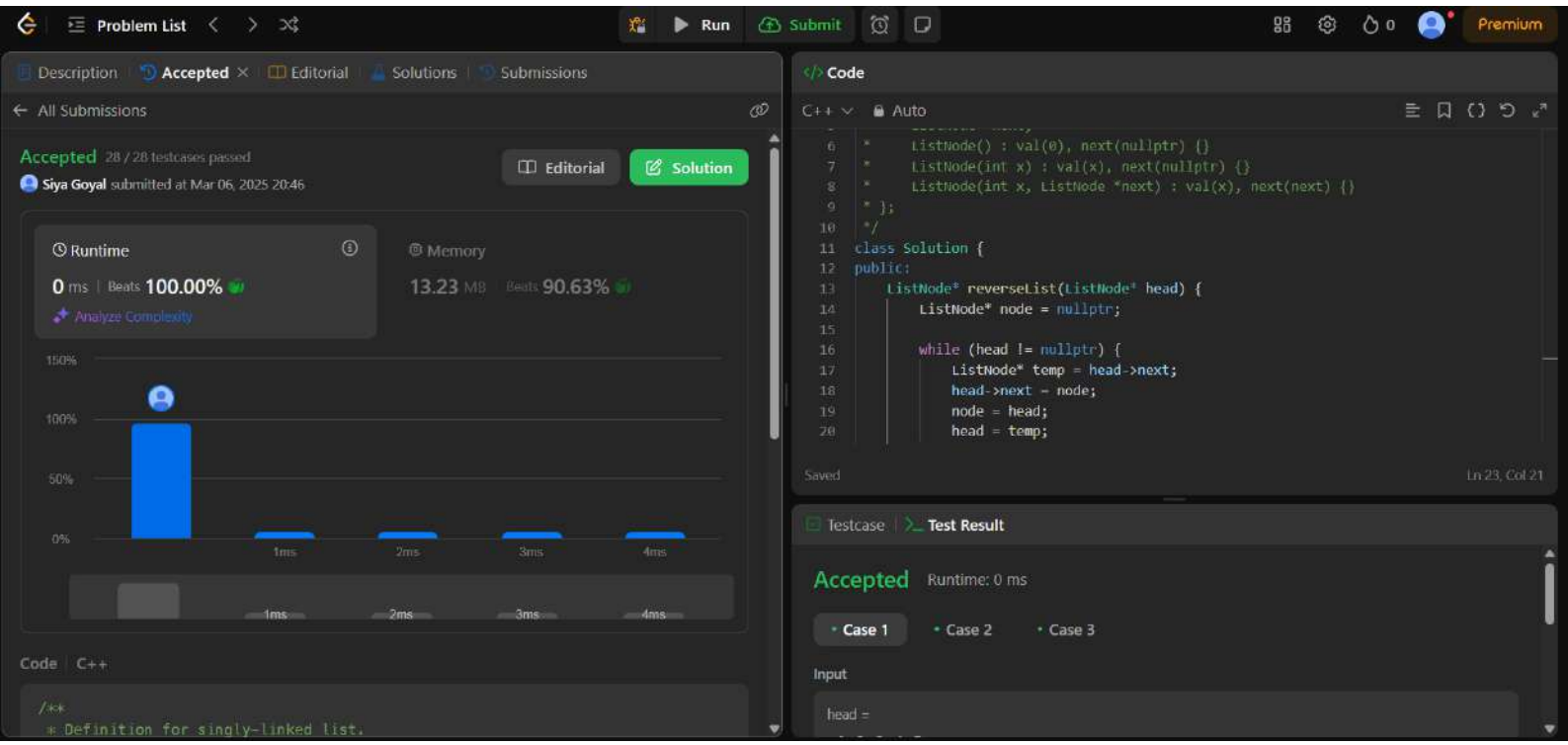
Input

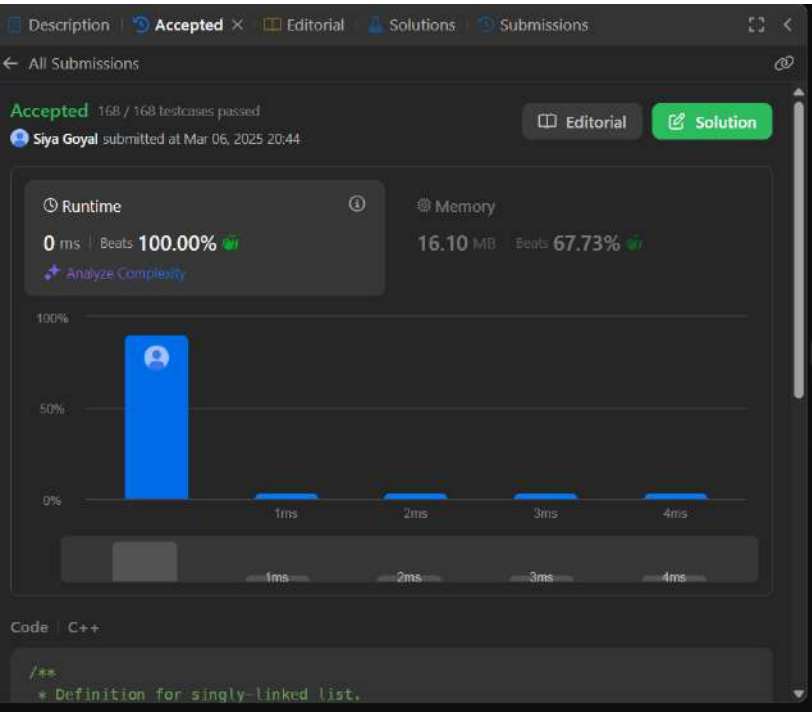
head =

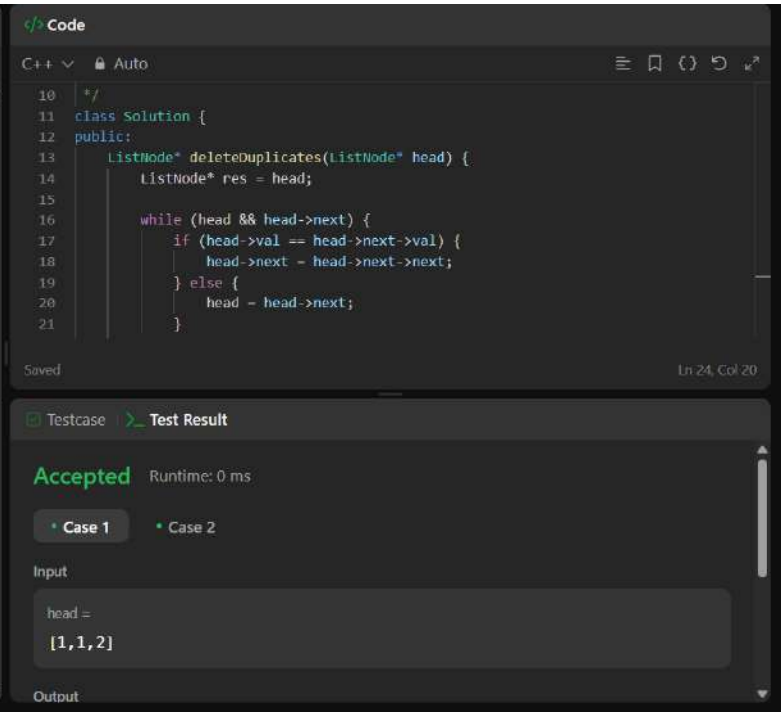
[3,2,0,-4]

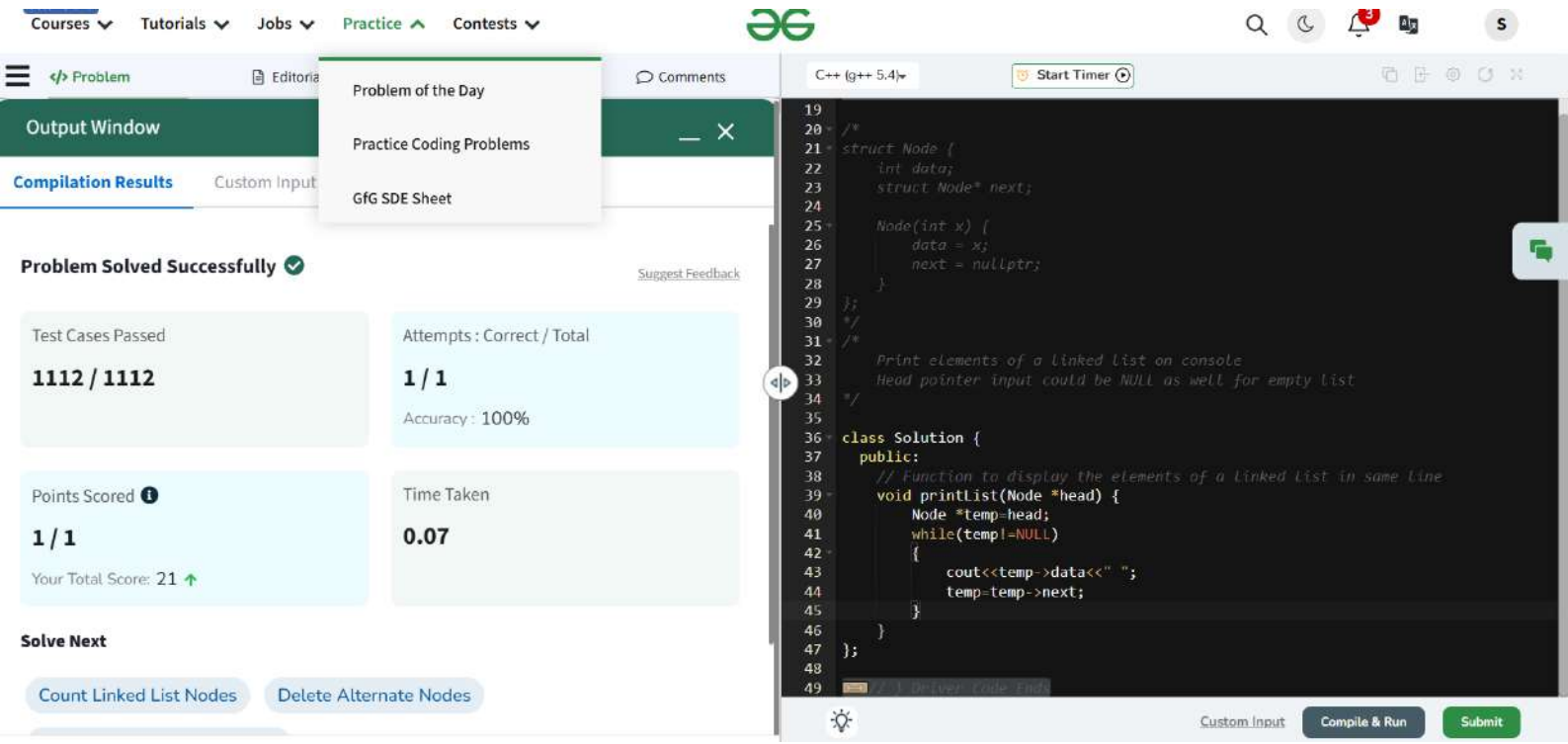

Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 208 / 208 testcases passed
Siya Goyal submitted at Mar 06, 2025 20:54
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
19.60 MB | Beats 28.39%
100%
50%
0%
1ms
2ms
3ms
4ms
Code | C++
/**
* Definition for singly-linked list.
Code
C++
Auto
class Solution {
public:
ListNode* mergeTwoLists(ListNode* list1, ListNode* list2) {
if(list1 == NULL)
return list2;
if(list2 == NULL)
return list1;
ListNode * ptr = list1;
if(list1 -> val > list2 -> val)
{
ptr = list2;
list2 = list2 -> next;
}
Saved
Ln 16, Col 2
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
list1 =
[1,2,4]


Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 70 / 70 testcases passed
Siya Goyal submitted at Mar 06, 2025 20:49
Editorial
Solution
Runtime
2 ms | Beats 50.07%
Analyze Complexity
Memory
312.10 MB | Beats 55.26%
60%
40%
20%
0%
1ms
2ms
3ms
4ms
5ms
6ms
7ms
Code | C++
/**
* Definition for singly-linked list.
Code
C++
Auto
class Solution {
public:
ListNode* deleteMiddle(ListNode* head) {
if(!head->next) return NULL;
if(!head->next->next){
head->next = NULL;
return head;
}
ListNode* slow = head;
ListNode* fast = head;
while(fast && fast->next){
slow = slow->next;
fast = fast->next->next;
}
Saved
Ln 27, Col 21
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
head =
[1,3,4,7,1,2,6]


Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 28 / 28 testcases passed
Siya Goyal submitted at Mar 06, 2025 20:46
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
13.23 MB Beats 90.63%
150%
100%
50%
0%
1ms
2ms
3ms
4ms
Code C++
/**
* Definition for singly-linked list.
Code
C++
Auto
* 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;
Saved
Ln 23, Col 21
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
head =


Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 168 / 168 testcases passed
Siya Goyal submitted at Mar 06, 2025 20:44
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
16.10 MB Beats 67.73%
100%
50%
0%
1ms
2ms
3ms
4ms
Code | C++
/**
* Definition for singly-linked list.


Code
C++
Auto
*/
class Solution {
public:
ListNode* deleteDuplicates(ListNode* head) {
ListNode* res = head;
while (head && head->next) {
if (head->val == head->next->val) {
head->next = head->next->next;
} else {
head = head->next;
}
Saved
Ln 24, Col 20
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
head =
[1,1,2]
Output


Courses
Tutorials
Jobs
Practice
Contests
Problem
Editorial
Comments
Problem of the Day
Practice Coding Problems
GfG SDE Sheet
Output Window
Compilation Results
Custom Input
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1112 / 1112
Attempts : Correct / Total
1 / 1
Accuracy : 100%
Points Scored
1 / 1
Your Total Score: 21
Time Taken
0.07
Solve Next
Count Linked List Nodes
Delete Alternate Nodes
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) {
Node *temp=head;
while(temp!=NULL)
{
cout<<temp->data<<" ";
temp=temp->next;
}
}
};
Custom Input
Compile & Run
Submit