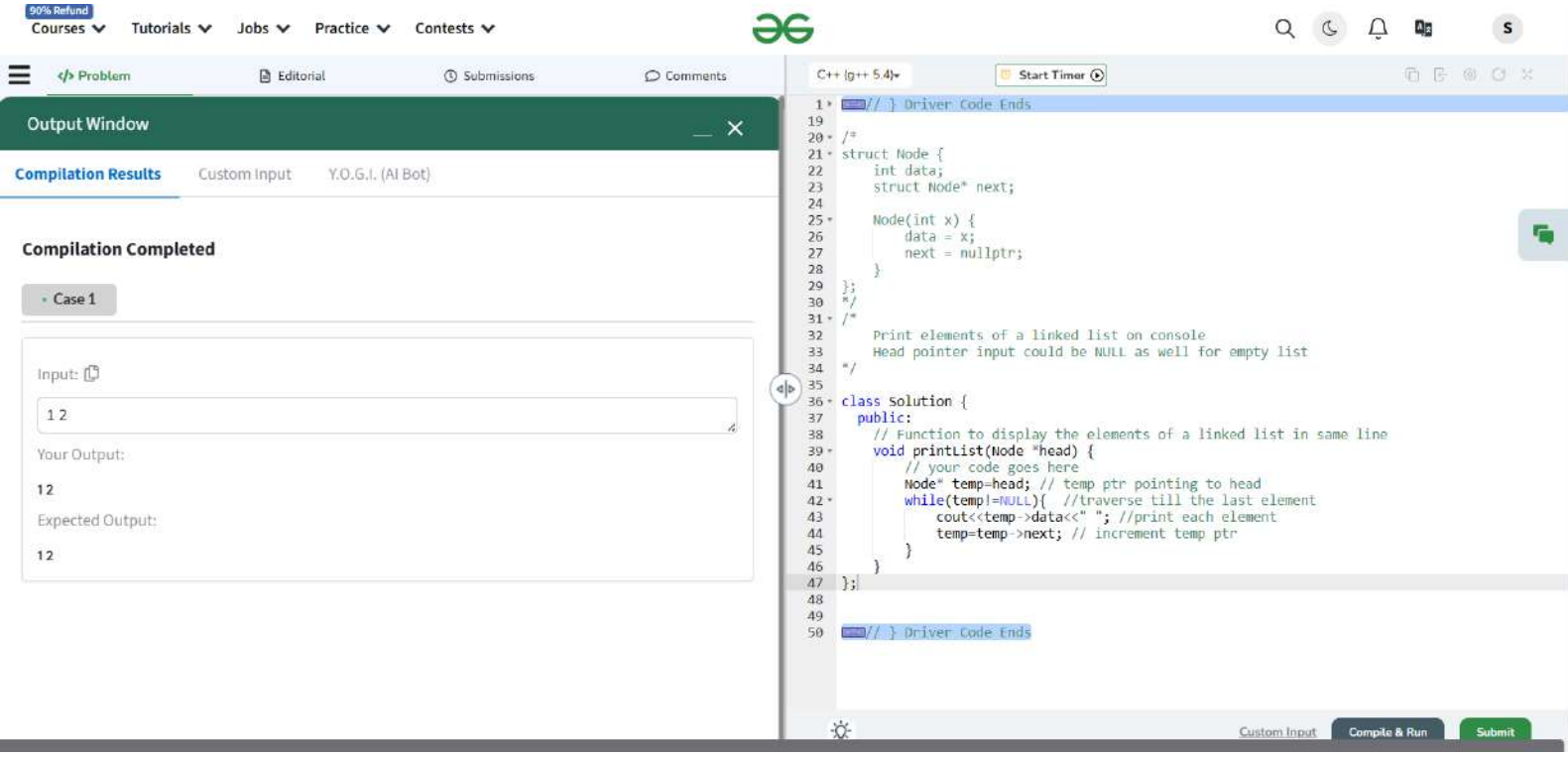
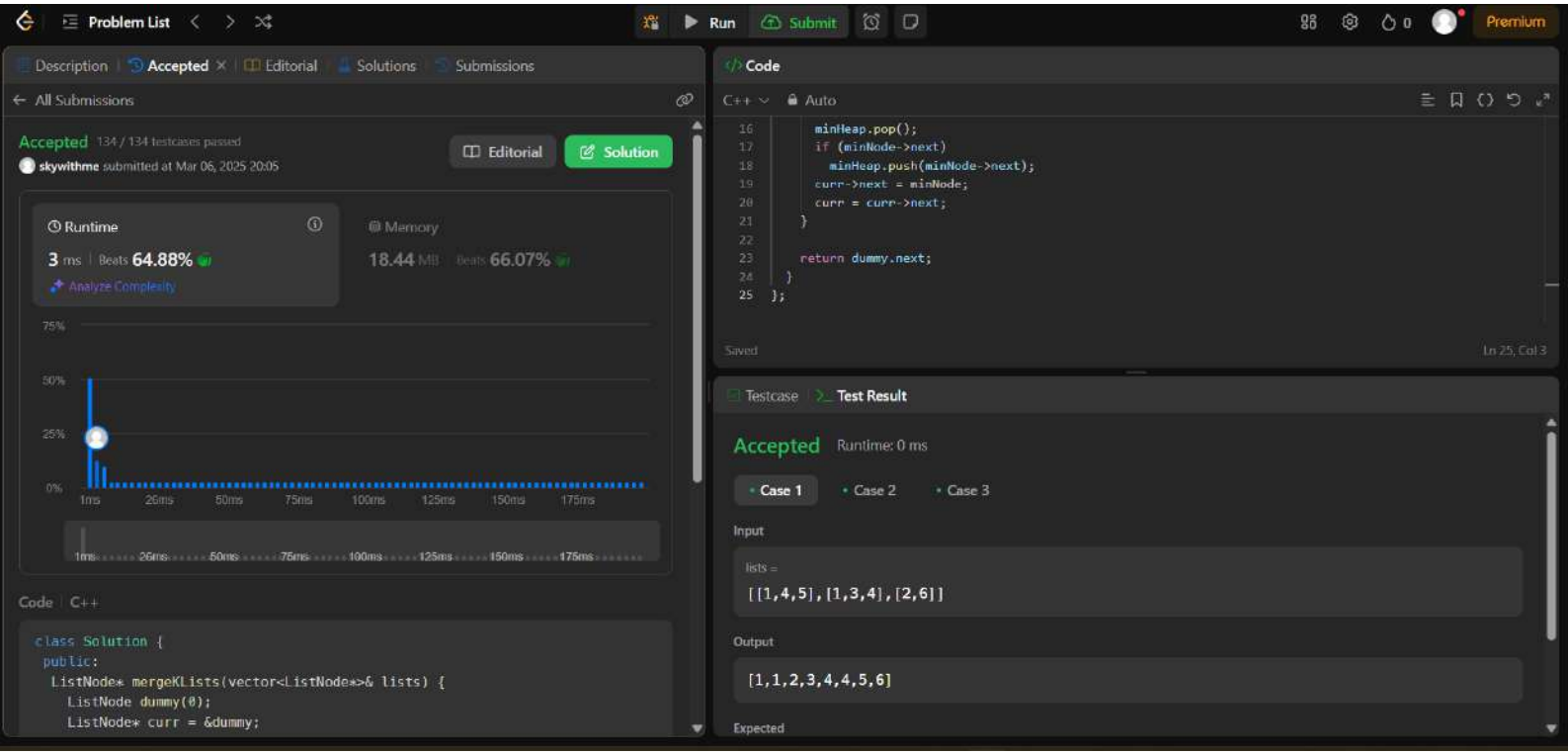
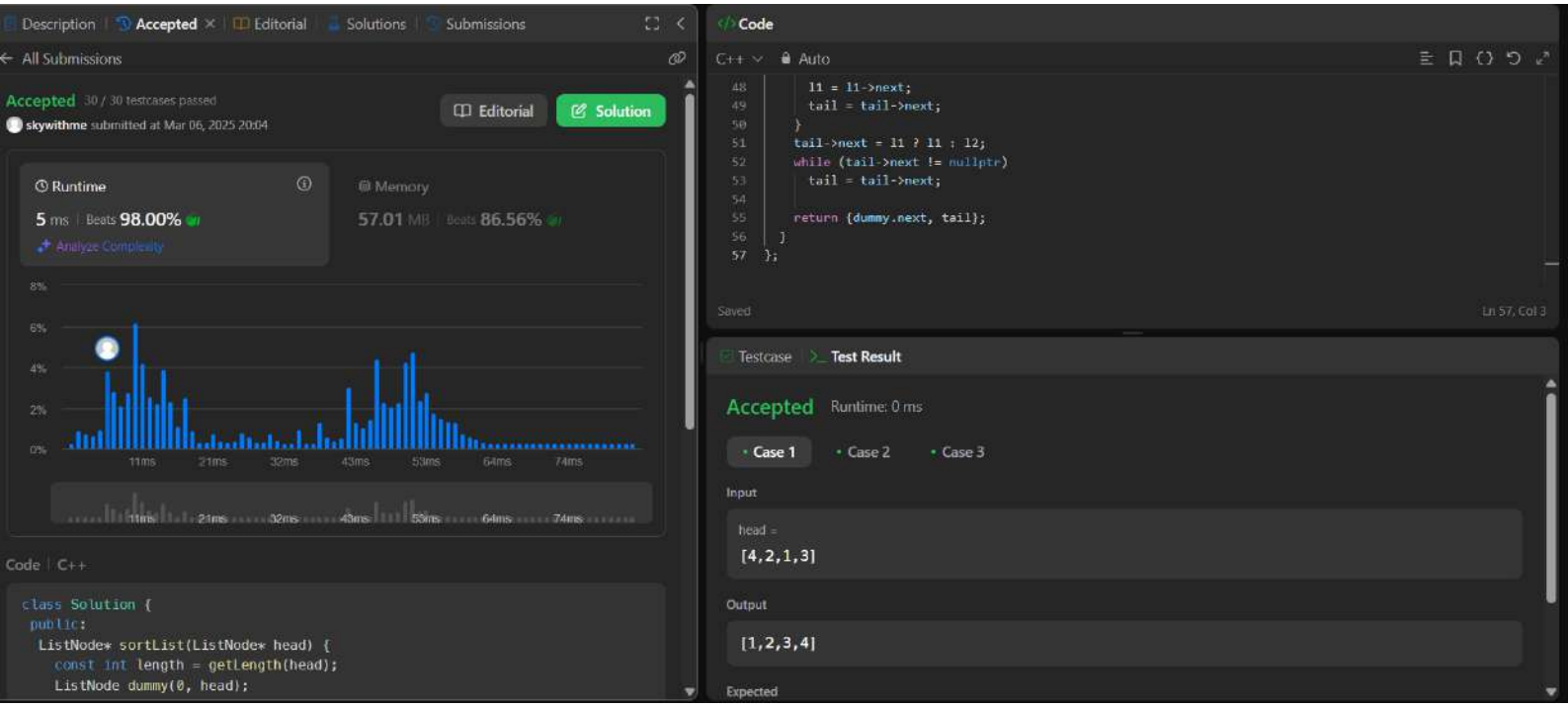

90% Refund
Courses
Tutorials
Jobs
Practice
Contests
Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Compilation Completed
Case 1
Input:
1 2
Your Output:
1 2
Expected Output:
1 2
C++ (g++ 5.4)
Start Timer
// } Driver Code Ends
/*
struct Node {
int 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:
// Function to display the elements of a linked list in same line
void printList(Node *head) {
// your code goes here
Node* temp=head; // temp ptr pointing to head
while(temp!=NULL){ //traverse till the last element
cout<<temp->data<<" "; //print each element
temp=temp->next; // increment temp ptr
}
}
};
// } Driver Code Ends
Custom Input
Compile & Run
Submit


Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 134 / 134 testcases passed
skywithme submitted at Mar 06, 2025 20:05
Editorial
Solution
Runtime
3 ms | Beats 64.88%
Analyze Complexity
Memory
18.44 MB Beats 66.07%
75%
50%
25%
0%
1ms
26ms
50ms
75ms
100ms
125ms
150ms
175ms
Code | C++
class Solution {
public:
ListNode* mergeKLists(vector<ListNode*>& lists) {
ListNode dummy(0);
ListNode* curr = &dummy;
Code
C++
Auto
minHeap.pop();
if (minNode->next)
minHeap.push(minNode->next);
curr->next = minNode;
curr = curr->next;
}
return dummy.next;
}
};
Saved
Ln 25, Col 3
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


Description | Accepted | Editorial | Solutions | Submissions
All Submissions
Accepted 30 / 30 testcases passed
skywithme submitted at Mar 06, 2025 20:04
Editorial
Solution
Runtime
5 ms | Beats 98.00%
Analyze Complexity
Memory
57.01 MB | Beats 86.56%
8%
6%
4%
2%
0%
11ms
21ms
32ms
43ms
53ms
64ms
74ms
Code | C++
class Solution {
public:
ListNode* sortList(ListNode* head) {
const int length = getLength(head);
ListNode dummy(0, head);
Code
C++
Auto
11 = 11->next;
tail = tail->next;
}
tail->next = 11 ? 11 : 12;
while (tail->next != nullptr)
tail = tail->next;
return {dummy.next, tail};
}
};
Saved
Ln 57, Col 3
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

Description | Accepted × | Editorial | Solutions | Submissions

← All Submissions

**Accepted** 232 / 232 testcases passed

skywithme submitted at Mar 06, 2025 20:03

Editorial | Solution

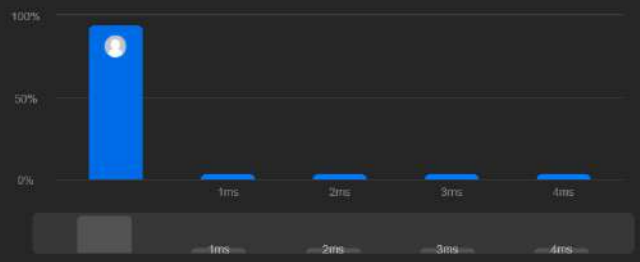
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

16.38 MB Beats 64.71%




Code | C++

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

Code

C++ | Auto

```cpp
12
13    const int t = length - k % length;
14    for (int i = 0; i < t; ++i)
15      tail = tail->next;
16    ListNode* newHead = tail->next;
17    tail->next = nullptr;
18
19    return newHead;
20  }
21 };
```

Saved

Ln 21, Col 3

Testcase | Test Result

**Accepted** Runtime: 0 ms

• Case 1 • Case 2

Input

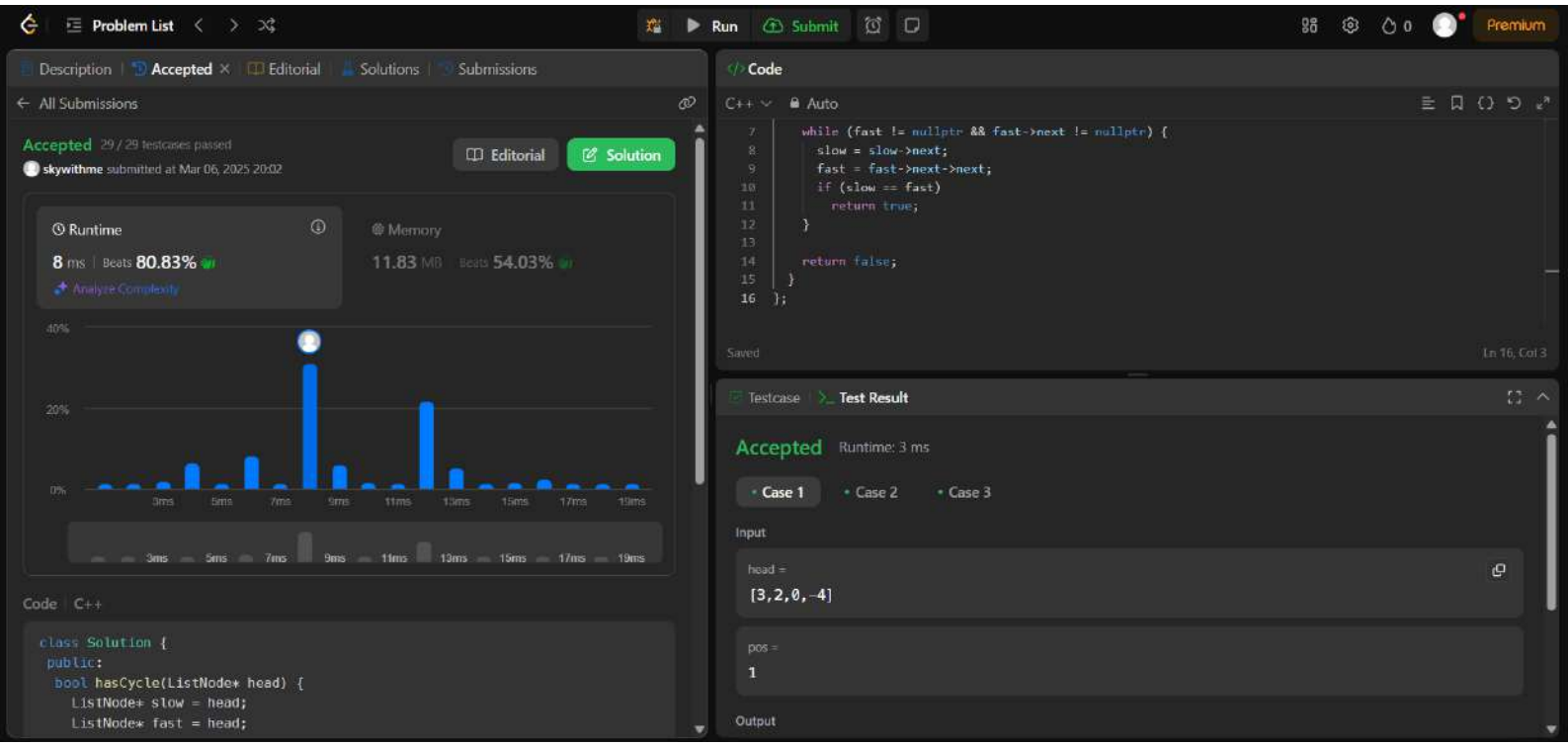
head =

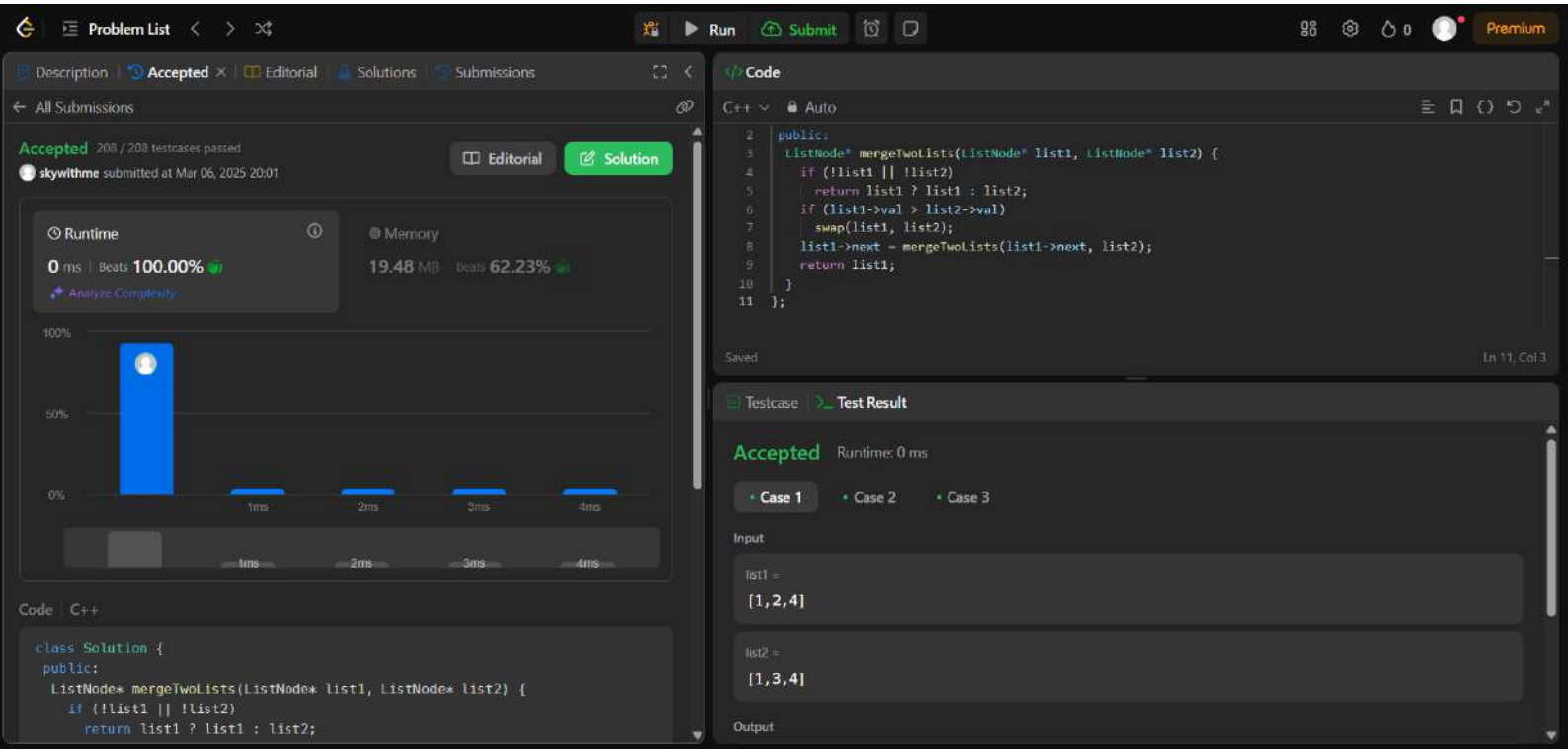
[1,2,3,4,5]

k =

2

Output


Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 29 / 29 testcases passed
skywithme submitted at Mar 06, 2025 20:02
Editorial
Solution
Runtime
8 ms | Beats 80.83%
Analyze Complexity
Memory
11.83 MB Beats 54.03%
40%
20%
0%
3ms 5ms 7ms 9ms 11ms 13ms 15ms 17ms 19ms
Code | C++
class Solution {
public:
bool hasCycle(ListNode* head) {
ListNode* slow = head;
ListNode* fast = head;
Code
C++
Auto
while (fast != nullptr && fast->next != nullptr) {
slow = slow->next;
fast = fast->next->next;
if (slow == fast)
return true;
}
return false;
}
};
Saved
Ln 16, Col 3
Testcase
Test Result
Accepted
Runtime: 3 ms
Case 1
Case 2
Case 3
Input
head =
[3,2,0,-4]
pos =
1
Output


Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 208 / 208 testcases passed
skywithme submitted at Mar 06, 2025 20:01
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
19.48 MB | Beats 62.23%
100%
50%
0%
1ms
2ms
3ms
4ms
Code | C++
class Solution {
public:
ListNode* mergeTwoLists(ListNode* list1, ListNode* list2) {
if (!list1 || !list2)
return list1 ? list1 : list2;
Code
C++
Auto
public:
ListNode* mergeTwoLists(ListNode* list1, ListNode* list2) {
if (!list1 || !list2)
return list1 ? list1 : list2;
if (list1->val > list2->val)
swap(list1, list2);
list1->next = mergeTwoLists(list1->next, list2);
return list1;
}
};
Saved
Ln 11, Col 3
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
list1 =
[1,2,4]
list2 =
[1,3,4]
Output

Problem List | Run | Submit | Premium

Description | Editorial | Solutions | Submissions

# 2095. Delete the Middle Node of a Linked List

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]
Output: [1,3,4,1,2,6]
Explanation:
The above figure represents the given linked list. The indices of the
nodes are written below.
Since n = 7, node 3 with value 7 is the middle node, which is marked in
red.
We return the new list after removing this node.
```

4.5K | 77 | 75 Online

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

Code

C++ | Auto

```
8     while (fast->next != nullptr && fast->next->next != nullptr) {
9       slow = slow->next;
10      fast = fast->next->next;
11    }
12
13    // Delete the middle node.
14    slow->next = slow->next->next;
15    return dummy.next;
16  }
17 };
```

Saved | Ln 17, Col 3

Testcase | Test Result

**Accepted** Runtime: 0 ms

Case 1 | Case 2 | Case 3

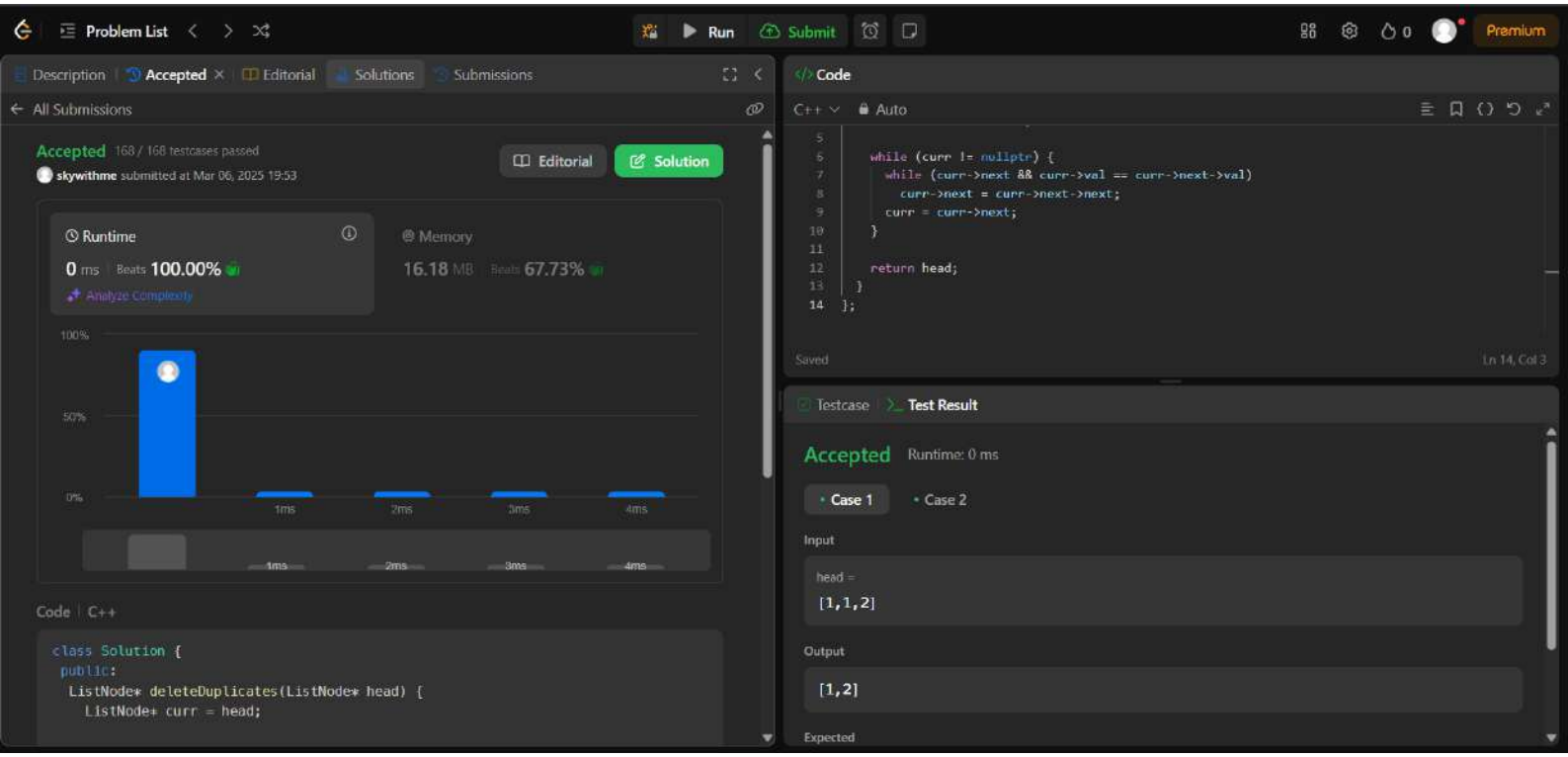
Input

```
head =
[1,3,4,7,1,2,6]
```

Output

```
[1,3,4,1,2,6]
```

Expected


Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 168 / 168 testcases passed
skywithme submitted at Mar 06, 2025 19:53
Editorial
Solution
Runtime
0 ms Beats 100.00%
Analyze Complexity
Memory
16.18 MB Beats 67.73%
100%
50%
0%
1ms
2ms
3ms
4ms
Code | C++
class Solution {
public:
ListNode* deleteDuplicates(ListNode* head) {
ListNode* curr = head;
Code
C++
Auto
while (curr != nullptr) {
while (curr->next && curr->val == curr->next->val)
curr->next = curr->next->next;
curr = curr->next;
}
return head;
}
};
Saved
Ln 14, Col 3
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
head =
[1,1,2]
Output
[1,2]
Expected