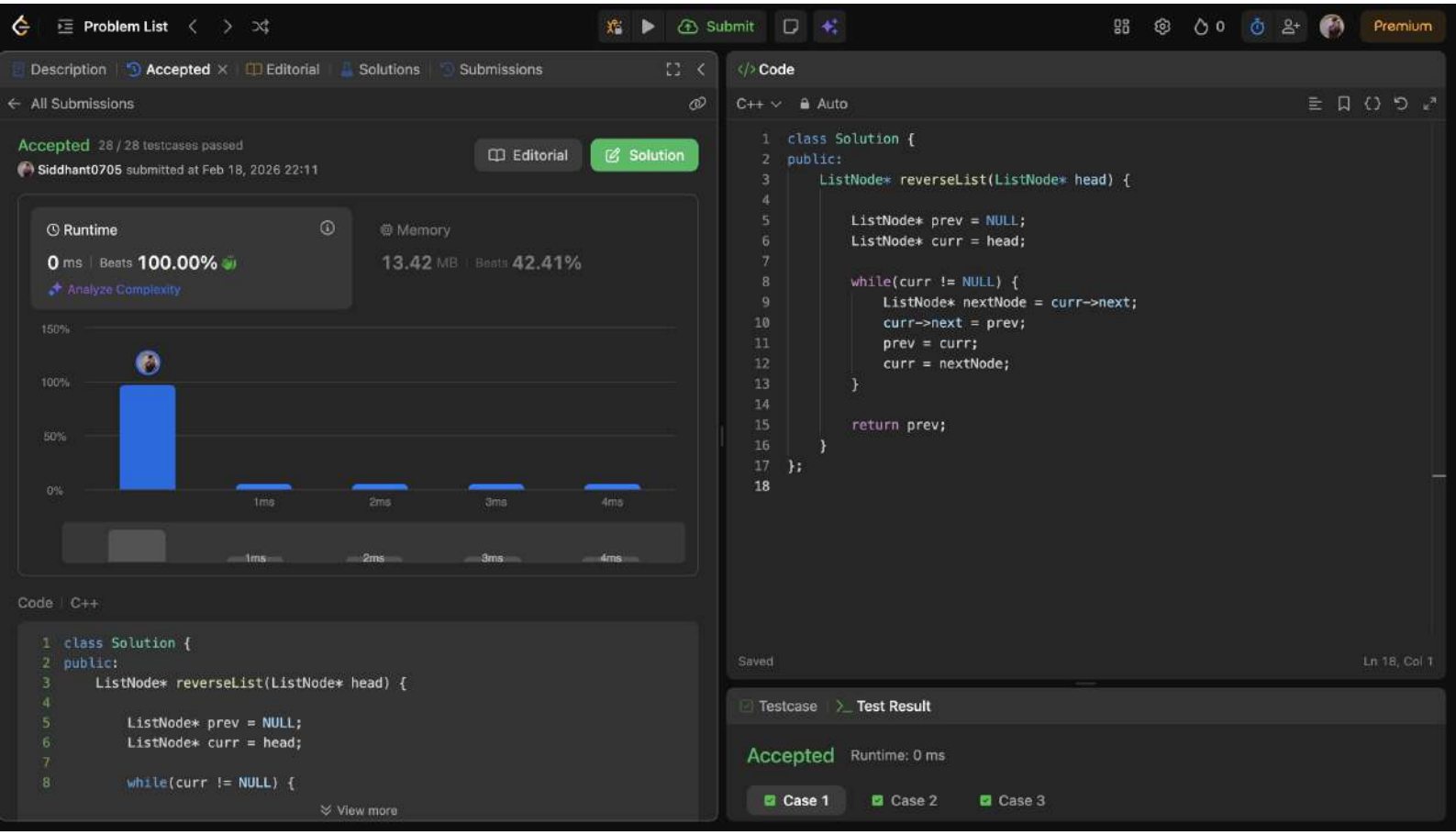
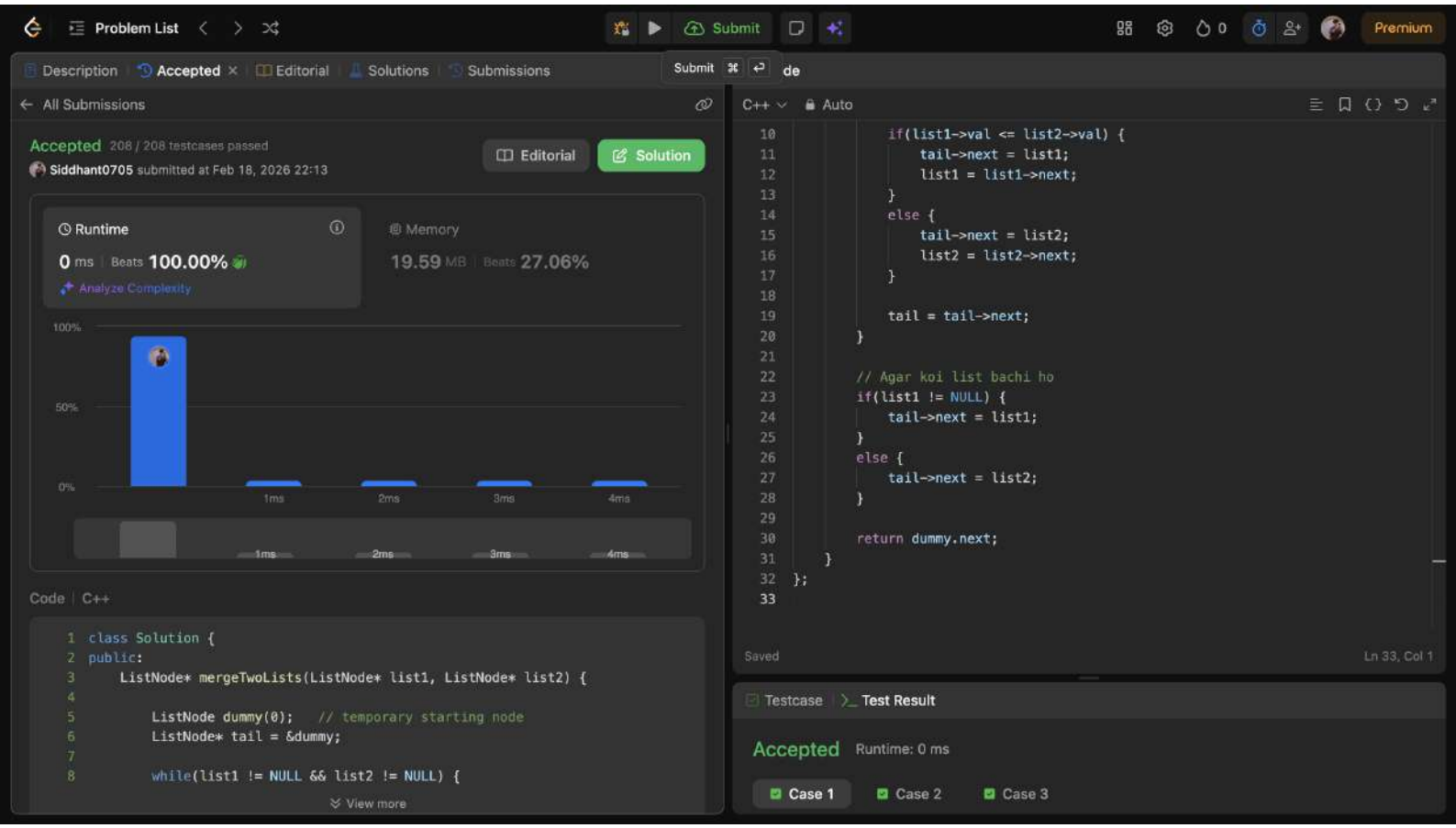

Problem List
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 28 / 28 testcases passed
Siddhant0705 submitted at Feb 18, 2026 22:11
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
13.42 MB | Beats 42.41%
150%
100%
50%
0%
1ms
2ms
3ms
4ms
Code | C++
class Solution {
public:
ListNode* reverseList(ListNode* head) {
ListNode* prev = NULL;
ListNode* curr = head;
while(curr != NULL) {
View more
Code
C++
Auto
class Solution {
public:
ListNode* reverseList(ListNode* head) {
ListNode* prev = NULL;
ListNode* curr = head;
while(curr != NULL) {
ListNode* nextNode = curr->next;
curr->next = prev;
prev = curr;
curr = nextNode;
}
return prev;
}
};
Saved
Ln 18, Col 1
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3


Problem List
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
Submit ⌘ ↵
de
All Submissions
Accepted 208 / 208 testcases passed
Siddhant0705 submitted at Feb 18, 2026 22:13
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
19.59 MB | Beats 27.06%
100%
50%
0%
1ms
2ms
3ms
4ms
Code | C++
1 class Solution {
2 public:
3 ListNode* mergeTwoLists(ListNode* list1, ListNode* list2) {
4
5 ListNode dummy(0); // temporary starting node
6 ListNode* tail = &dummy;
7
8 while(list1 != NULL && list2 != NULL) {
View more
C++
Auto
10 if(list1->val <= list2->val) {
11 tail->next = list1;
12 list1 = list1->next;
13 }
14 else {
15 tail->next = list2;
16 list2 = list2->next;
17 }
18
19 tail = tail->next;
20 }
21
22 // Agar koi list bachi ho
23 if(list1 != NULL) {
24 tail->next = list1;
25 }
26 else {
27 tail->next = list2;
28 }
29
30 return dummy.next;
31 }
32 };
33
Saved
Ln 33, Col 1
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3