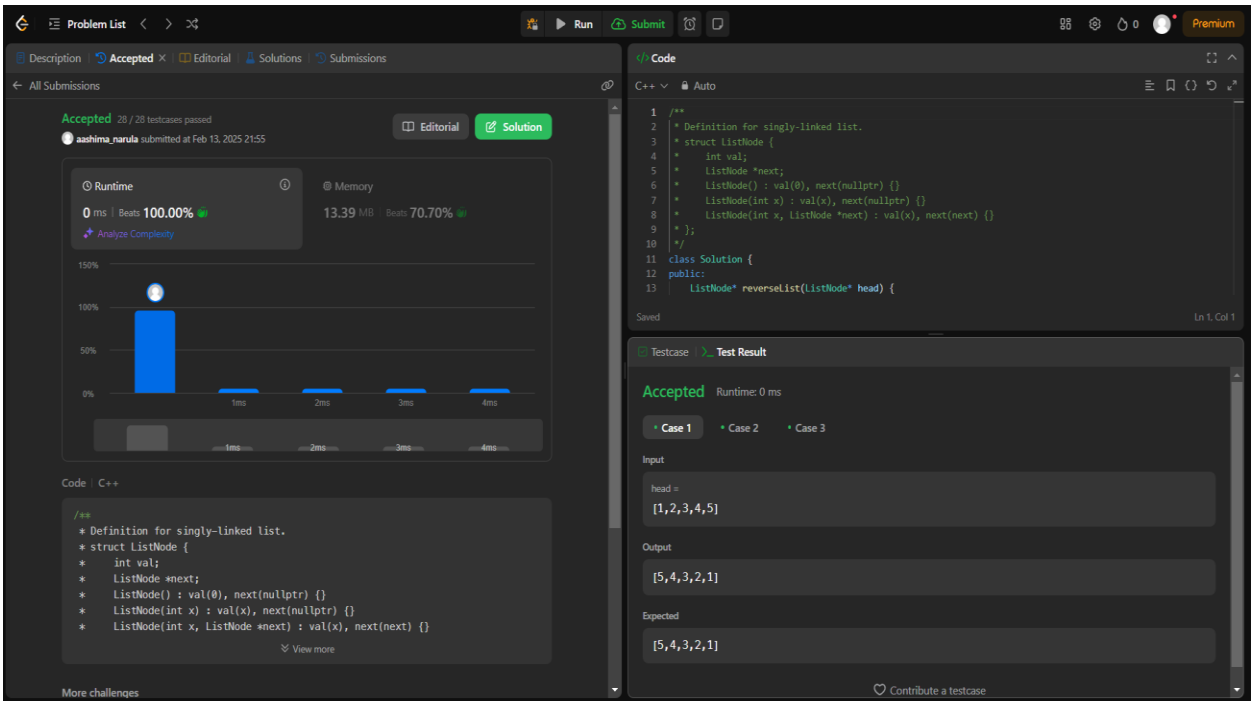
206. Reverse a Linked List

```
class Solution {
public:
    ListNode* reverseList(ListNode* head) {
        ListNode* prev = nullptr;
        ListNode* curr = head;
    while (curr) {
        ListNode* nextNode = curr->next;
        curr->next = prev;
        prev = curr;
        curr = nextNode;
    }
    return prev;
    }
};
```


Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 28 / 28 testcases passed
aashima_narula submitted at Feb 13, 2025 21:55
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
13.39 MB | Beats 70.70%
150%
100%
50%
0%
1ms
2ms
3ms
4ms
Code | C++
/**
* Definition for singly-linked list.
* struct ListNode {
*   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) {}
View more
More challenges
Code
C++
Auto
1 /**
2 * Definition for singly-linked list.
3 * struct ListNode {
4 *   int val;
5 *   ListNode *next;
6 *   ListNode() : val(0), next(nullptr) {}
7 *   ListNode(int x) : val(x), next(nullptr) {}
8 *   ListNode(int x, ListNode *next) : val(x), next(next) {}
9 * };
10 */
11 class Solution {
12 public:
13   ListNode* reverseList(ListNode* head) {
Saved
Ln 1, Col 1
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