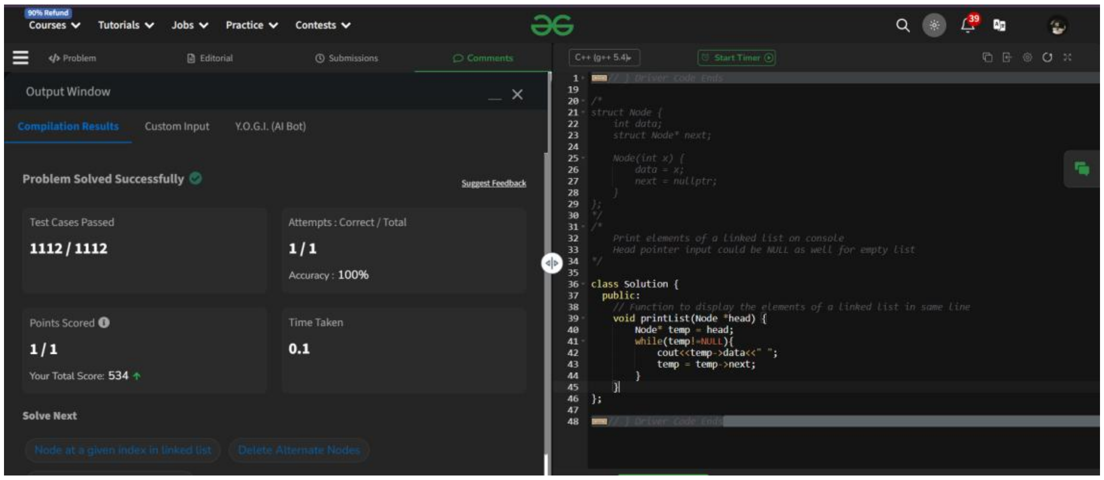
NAME – YASH GARG

UID – 22BCS13420

22BCS_IOT_608-B

1. Print Linked List: https://www.geeksforgeeks.org/problems/print-linked-list-elements/0



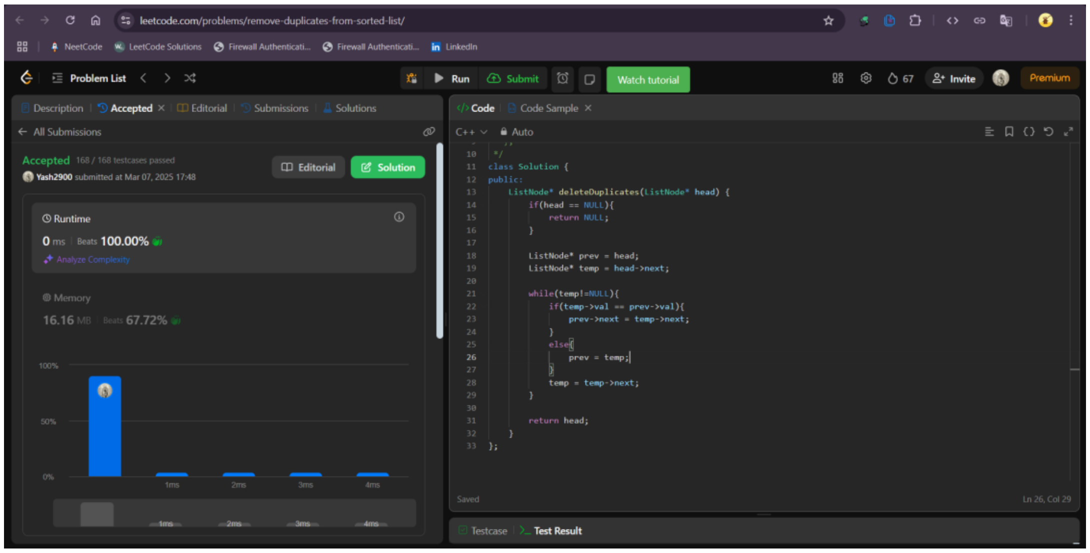
Output Window

Problem Solved Successfully

Test Cases Passed: 1112 / 1112
Attempts : Correct / Total: 1 / 1
Accuracy : 100%
Points Scored: 1 / 1
Your Total Score: 534
Time Taken: 0.1

```cpp
// } 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) {
        Node* temp = head;
        while(temp!=NULL){
            cout<<temp->data<<" ";
            temp = temp->next;
        }
    }
};

// } Driver Code Ends
```

2. Remove duplicates from a sorted list: https://leetcode.com/problems/remove-duplicates-from-sorted-list/description/

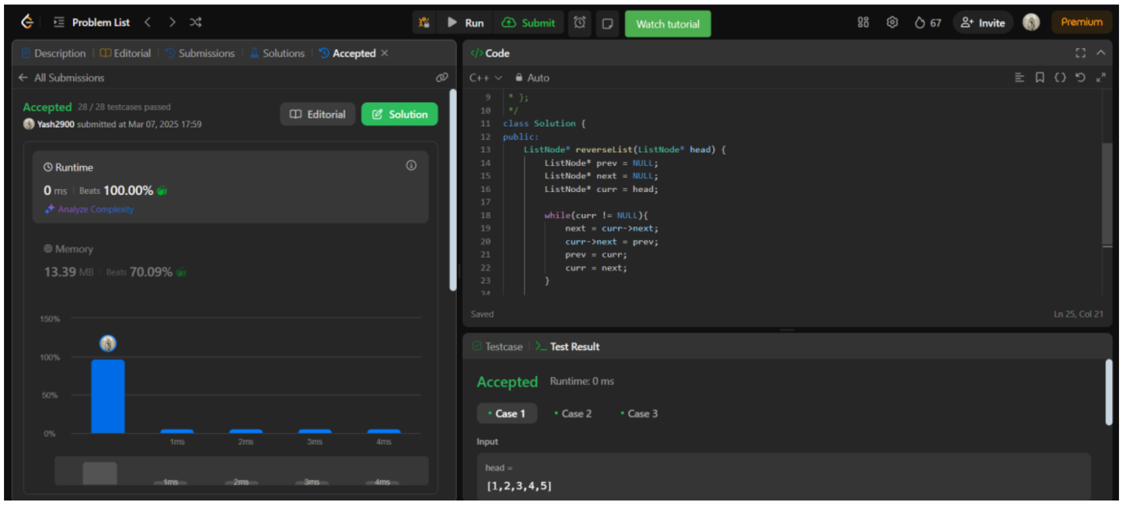


Accepted 168 / 168 testcases passed
Yash2900 submitted at Mar 07, 2025 17:48

Runtime: 0 ms | Beats 100.00%
Memory: 16.16 MB | Beats 67.72%

```cpp
 */
class Solution {
public:
    ListNode* deleteDuplicates(ListNode* head) {
        if(head == NULL){
            return NULL;
        }

        ListNode* prev = head;
        ListNode* temp = head->next;

        while(temp!=NULL){
            if(temp->val == prev->val){
                prev->next = temp->next;
            }
            else{
                prev = temp;
            }
            temp = temp->next;
        }

        return head;
    }
};
```

3. Reverse a linked list: https://leetcode.com/problems/reverse-linked-list/description/



Accepted 28 / 28 testcases passed
Yash2900 submitted at Mar 07, 2025 17:59

Runtime: 0 ms | Beats 100.00%
Memory: 13.39 MB | Beats 70.09%

```cpp
 * };
 */
class Solution {
public:
    ListNode* reverseList(ListNode* head) {
        ListNode* prev = NULL;
        ListNode* next = NULL;
        ListNode* curr = head;

        while(curr != NULL){
            next = curr->next;
            curr->next = prev;
            prev = curr;
            curr = next;
        }
```

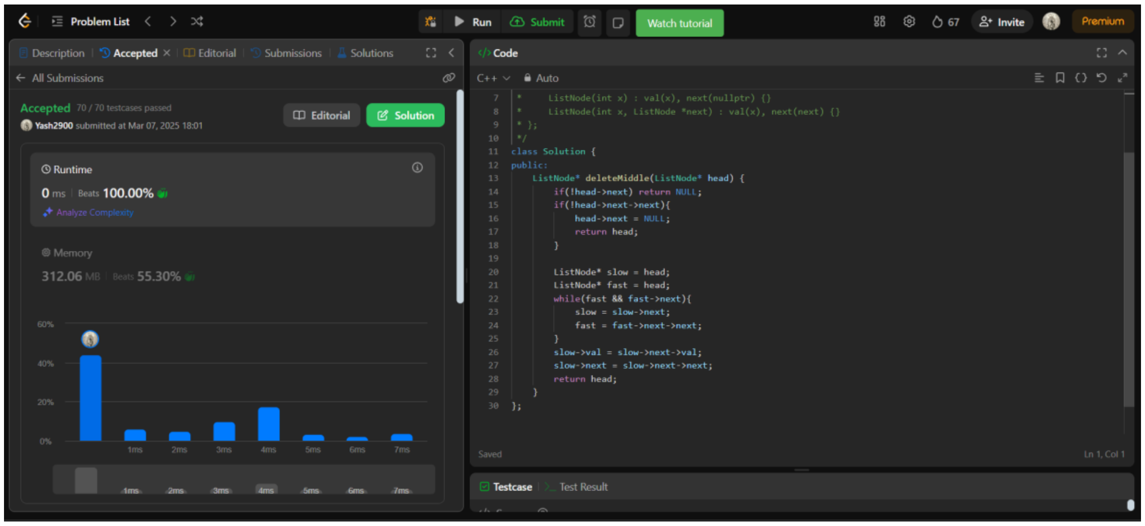
Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

head =
[1,2,3,4,5]

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

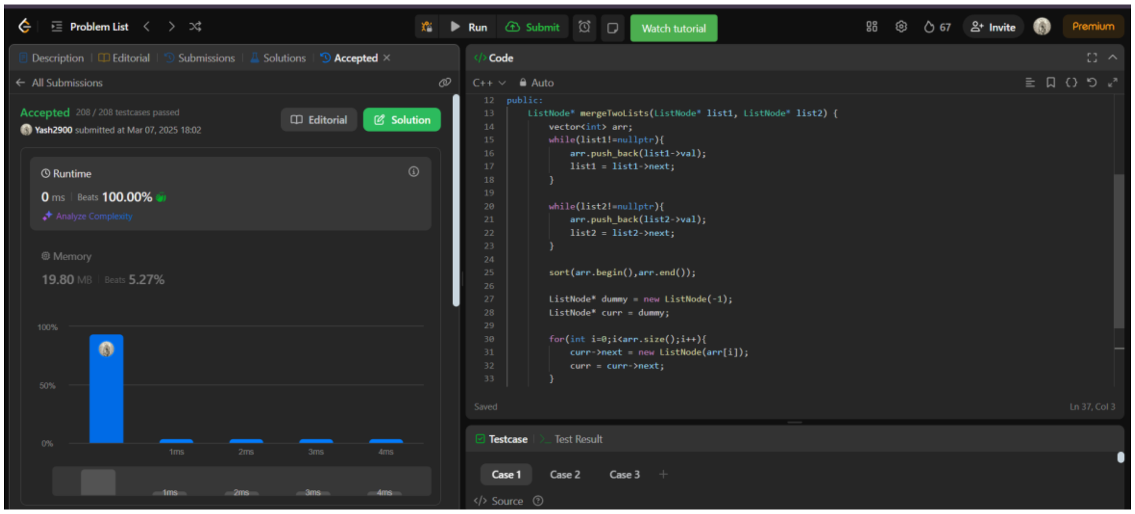

Accepted 70 / 70 testcases passed
Yash2900 submitted at Mar 07, 2025 18:01

Runtime
0 ms | Beats 100.00%

Memory
312.06 MB | Beats 55.30%

```
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
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;
        }
        slow->val = slow->next->val;
        slow->next = slow->next->next;
        return head;
    }
};
```

5. Merge two sorted linked lists: https://leetcode.com/problems/merge-two-sorted-lists/description/

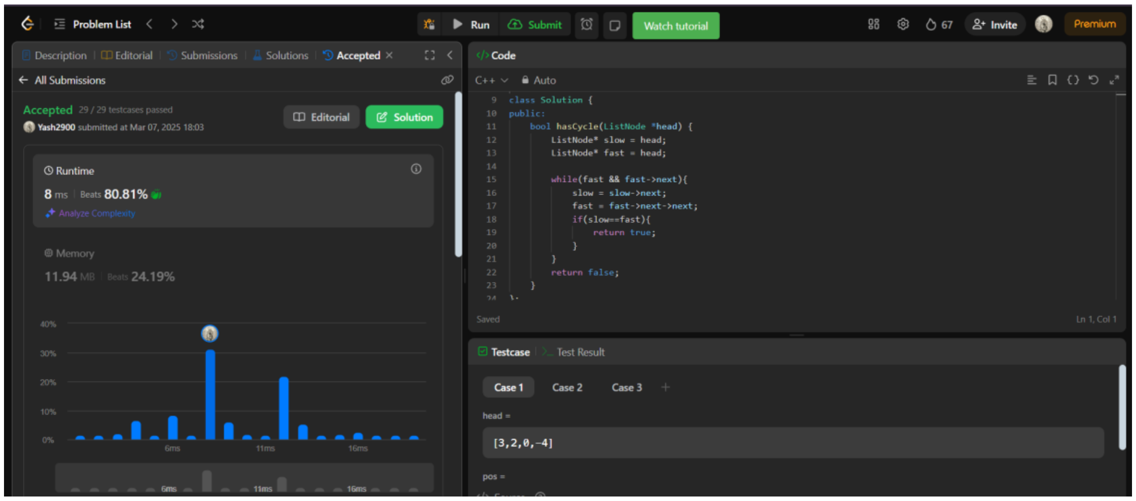

Accepted 208 / 208 testcases passed
Yash2900 submitted at Mar 07, 2025 18:02

Runtime
0 ms | Beats 100.00%

Memory
19.80 MB | Beats 5.27%

```
public:
    ListNode* mergeTwoLists(ListNode* list1, ListNode* list2) {
        vector<int> arr;
        while(list1!=nullptr){
            arr.push_back(list1->val);
            list1 = list1->next;
        }

        while(list2!=nullptr){
            arr.push_back(list2->val);
            list2 = list2->next;
        }

        sort(arr.begin(),arr.end());

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

        for(int i=0;i<arr.size();i++){
            curr->next = new ListNode(arr[i]);
            curr = curr->next;
        }
```

6. Detect a cycle in a linked list: https://leetcode.com/problems/linked-list-cycle/description/


Accepted 29 / 29 testcases passed
Yash2900 submitted at Mar 07, 2025 18:03

Runtime
8 ms | Beats 80.81%

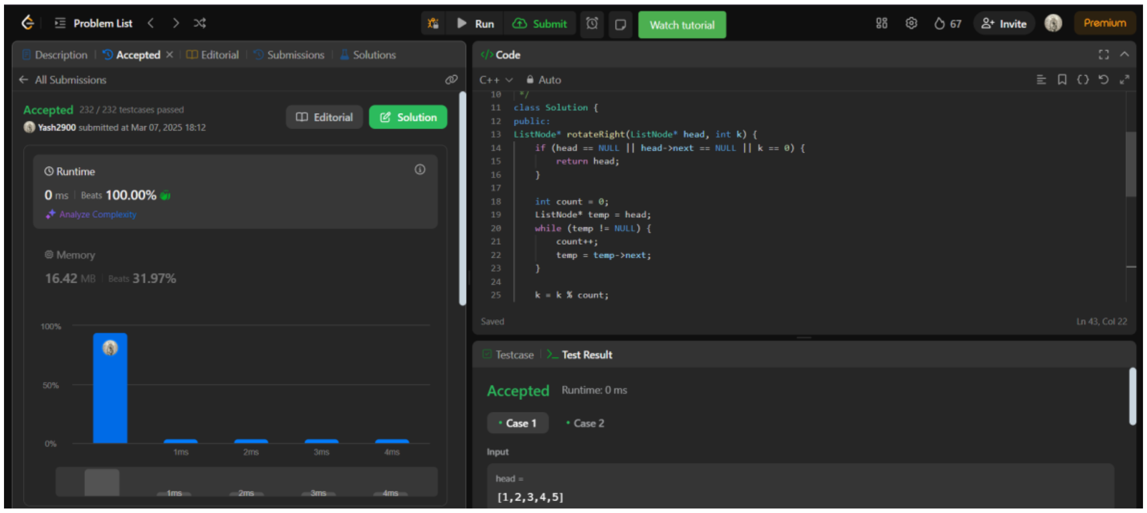
Memory
11.94 MB | Beats 24.19%

```
class Solution {
public:
    bool hasCycle(ListNode *head) {
        ListNode* slow = head;
        ListNode* fast = head;

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

head =
[3,2,0,-4]

pos =

7. Rotate a list: https://leetcode.com/problems/rotate-list/description/



Accepted 232 / 232 testcases passed

Yash2900 submitted at Mar 07, 2025 18:12

Runtime 0 ms Beats 100.00%

Memory 16.42 MB Beats 31.97%

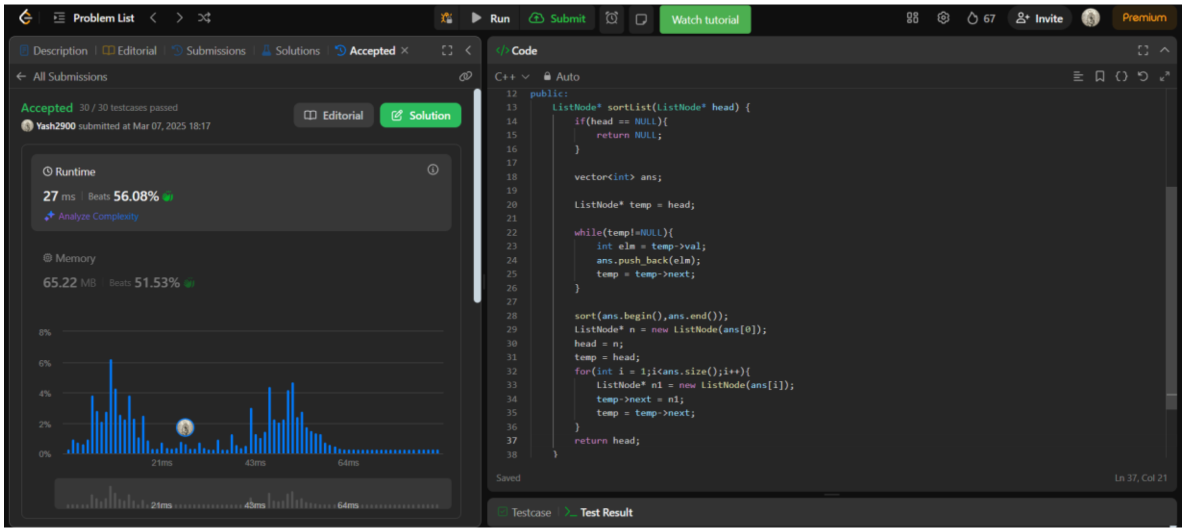
```cpp
 */
class Solution {
public:
ListNode* rotateRight(ListNode* head, int k) {
    if (head == NULL || head->next == NULL || k == 0) {
        return head;
    }

    int count = 0;
    ListNode* temp = head;
    while (temp != NULL) {
        count++;
        temp = temp->next;
    }

    k = k % count;
```

Accepted Runtime: 0 ms

Input

head =
[1,2,3,4,5]

8. Sort List: https://leetcode.com/problems/sort-list/description/



Accepted 30 / 30 testcases passed

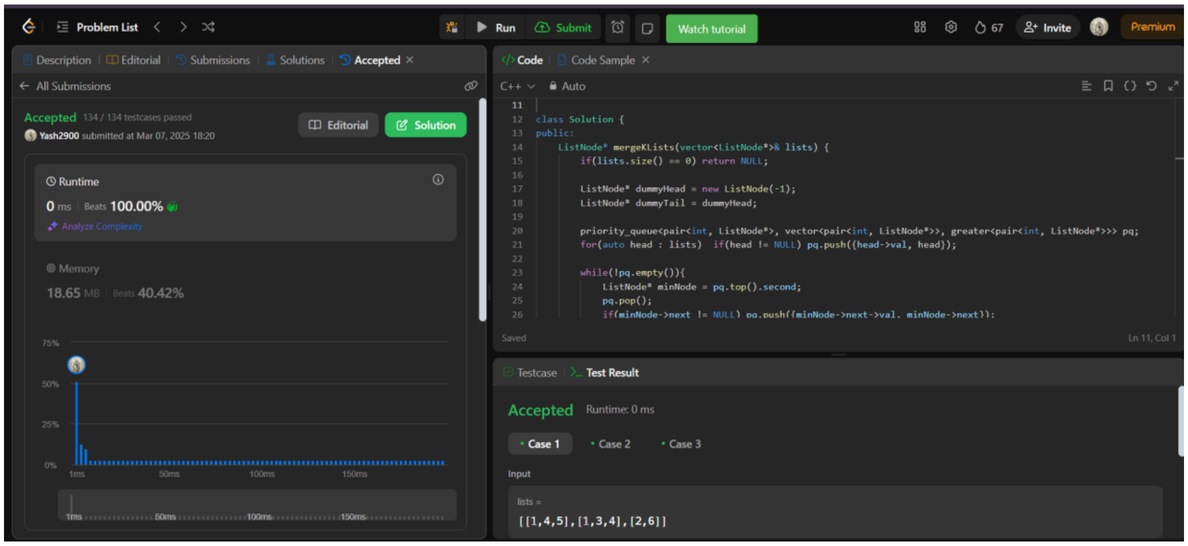
Yash2900 submitted at Mar 07, 2025 18:17

Runtime 27 ms Beats 56.08%

Memory 65.22 MB Beats 51.53%

```cpp
public:
    ListNode* sortList(ListNode* head) {
        if(head == NULL){
            return NULL;
        }

        vector<int> ans;

        ListNode* temp = head;

        while(temp!=NULL){
            int elm = temp->val;
            ans.push_back(elm);
            temp = temp->next;
        }

        sort(ans.begin(),ans.end());
        ListNode* n = new ListNode(ans[0]);
        head = n;
        temp = head;
        for(int i = 1;i<ans.size();i++){
            ListNode* n1 = new ListNode(ans[i]);
            temp->next = n1;
            temp = temp->next;
        }
        return head;
    }
```

9. Merge k sorted lists: https://leetcode.com/problems/merge-k-sorted-lists/description/



Accepted 134 / 134 testcases passed

Yash2900 submitted at Mar 07, 2025 18:20

Runtime 0 ms Beats 100.00%

Memory 18.65 MB Beats 40.42%

```cpp

class Solution {
public:
    ListNode* mergeKLists(vector<ListNode*>& lists) {
        if(lists.size() == 0) return NULL;

        ListNode* dummyHead = new ListNode(-1);
        ListNode* dummyTail = dummyHead;

        priority_queue<pair<int, ListNode*>, vector<pair<int, ListNode*>>, greater<pair<int, ListNode*>>> pq;
        for(auto head : lists)  if(head != NULL) pq.push({head->val, head});

        while(!pq.empty()){
            ListNode* minNode = pq.top().second;
            pq.pop();
            if(minNode->next != NULL) pq.push({minNode->next->val, minNode->next});
```

Accepted Runtime: 0 ms

Input

lists =
[[1,4,5],[1,3,4],[2,6]]