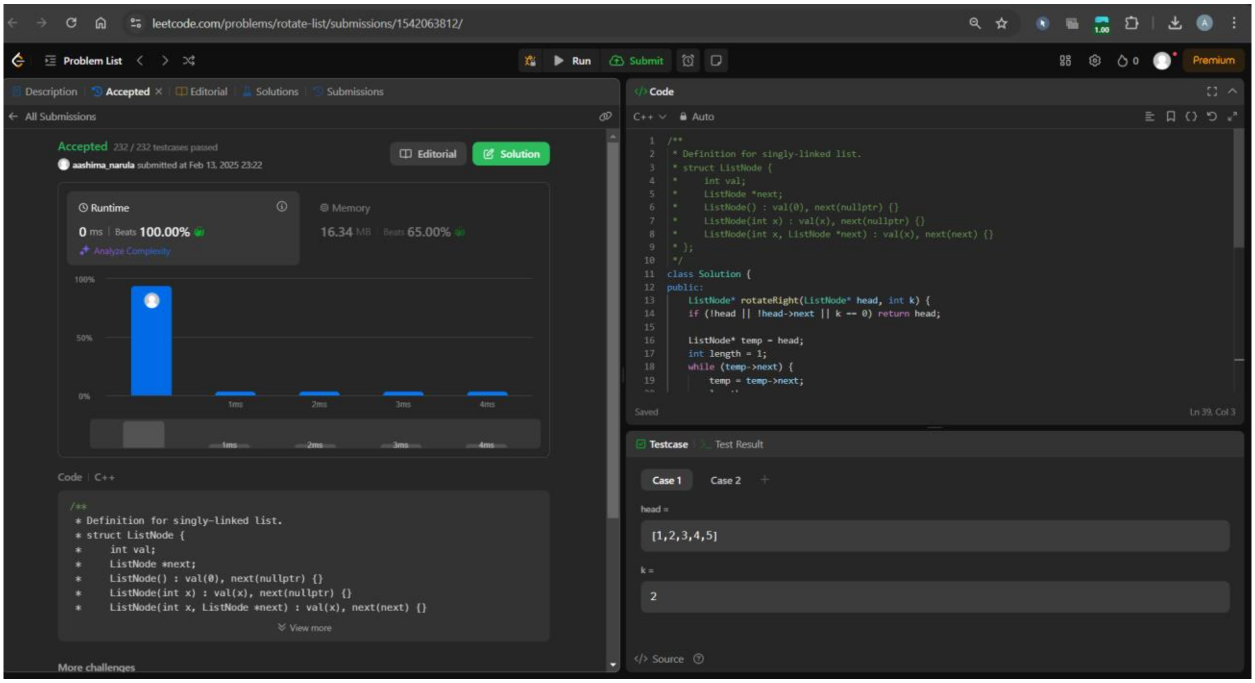
61. rotate a list

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

    ListNode* temp = head;
    int length = 1;
    while (temp->next) {
        temp = temp->next;
        length++;
    }

    k = k % length;
    if (k == 0) return head;

    ListNode* newTail = head;
    for (int i = 1; i < length - k; ++i) {
        newTail = newTail->next;
    }

    ListNode* newHead = newTail->next;

    temp->next = head;

    newTail->next = nullptr;
```

```
        return newHead;
    }
};
```



Accepted 232 / 232 testcases passed

aashima_narula submitted at Feb 13, 2025 23:22

Runtime: 0 ms | Beats 100.00%

Memory: 16.34 MB | Beats 65.00%

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) {}
```

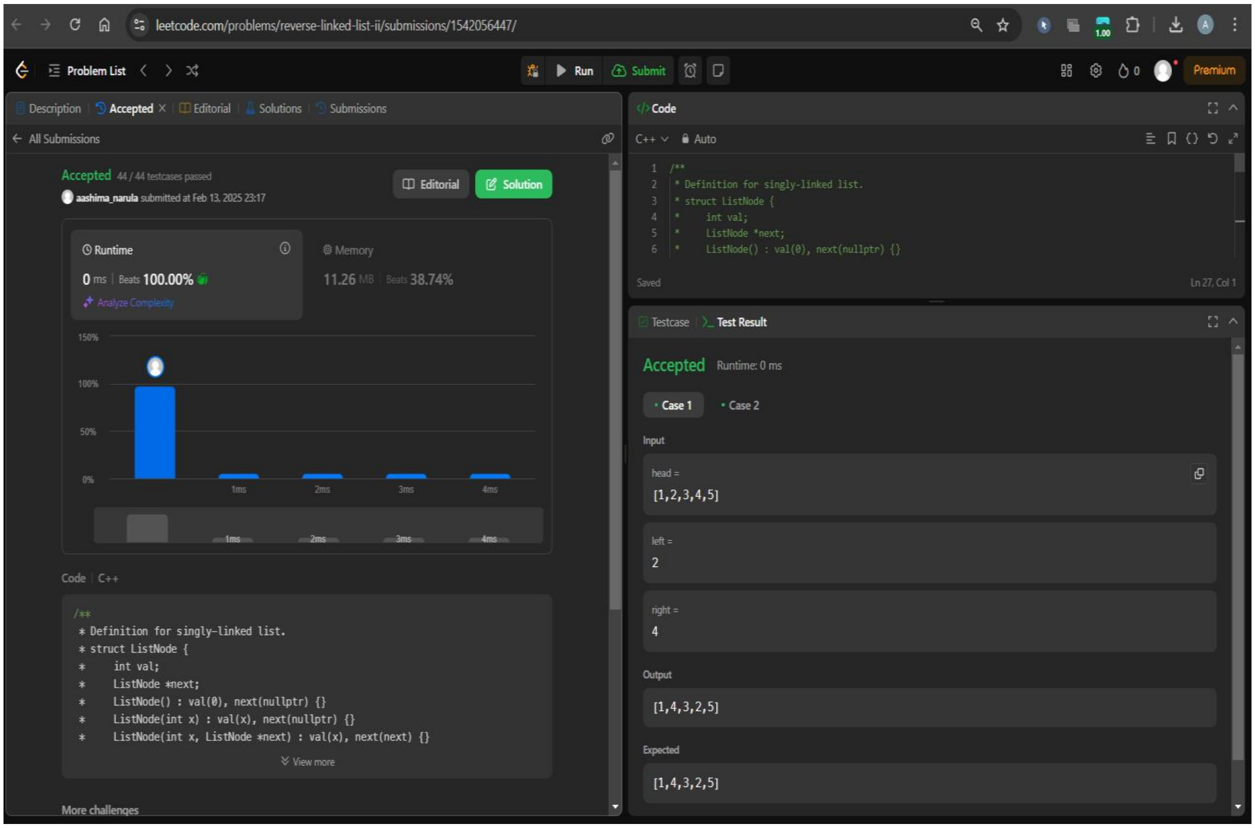
```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
class Solution {
public:
    ListNode* rotateRight(ListNode* head, int k) {
    if (!head || !head->next || k == 0) return head;

    ListNode* temp = head;
    int length = 1;
    while (temp->next) {
        temp = temp->next;
```

head =

```
[1,2,3,4,5]
```

k =

```
2
```



Accepted 44 / 44 testcases passed

aashima_narula submitted at Feb 13, 2025 23:17

Runtime

0 ms | Beats 100.00%

Memory

11.26 MB | Beats 38.74%

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) {}
```

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
```

Saved   Ln 27, Col 1

Accepted Runtime: 0 ms

Case 1   Case 2

Input

head =

[1,2,3,4,5]

left =

2

right =

4

Output

[1,4,3,2,5]

Expected

[1,4,3,2,5]