Assignment

Student Name: Gaurav Saini
UID: 22BCS11085
Branch: BE-CSE
Section/Group:NTPP_IOT_603_B
Semester:06
Date of Performance:04-04-2025

Subject Name: AP LAB-II
Subject Code: 22CSP-351

1. **Aim:**
   a. **Set Matrix Zero**
   b. **Reverse Linked List II**
   c. **Linked List Cycle**
   d. **Longest Substring Without Repeating Characters**

2. **Implementation/Code:**

A. Set Matrix Zero

```
class Solution {

public:

    void setZeroes(vector<vector<int>>& matrix) {

        int m = matrix.size();

        int n = matrix[0].size();



        bool firstRowZero = false;

        bool firstColZero = false;
```

```
for (int j = 0; j < n; ++j) {
    if (matrix[0][j] == 0) {
        firstRowZero = true;
        break;
    }
}

for (int i = 0; i < m; ++i) {
    if (matrix[i][0] == 0) {
        firstColZero = true;
        break;
    }
}

for (int i = 1; i < m; ++i) {
    for (int j = 1; j < n; ++j) {
        if (matrix[i][j] == 0) {
            matrix[i][0] = 0;
            matrix[0][j] = 0;
```

```
                }
            }
        }
        for (int i = 1; i < m; ++i) {
            for (int j = 1; j < n; ++j) {
                if (matrix[i][0] == 0 || matrix[0][j] == 0) {
                    matrix[i][j] = 0;
                }  }}
        if (firstRowZero) {
            for (int j = 0; j < n; ++j) {
                matrix[0][j] = 0;}}
        if (firstColZero) {
            for (int i = 0; i < m; ++i) {
                matrix[i][0] = 0;
            }
        }

    }
};
```

B. Reverse Linked List II

```
/**
 * 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* reverseBetween(ListNode* head, int left, int right) {
        if (!head || left == right) return head;

        ListNode* dummy = new ListNode(0);
        dummy->next = head;

        ListNode* prev = dummy;



        for (int i = 1; i < left; i++) {

            prev = prev->next;

        }



        ListNode* start = prev->next;



        ListNode* then = start->next;



        for (int i = 0; i < right - left; i++) {

            start->next = then->next;

            then->next = prev->next;

            prev->next = then;

            then = start->next;

        }

    return dummy->next;}

};
```

C. Linked List Cycle

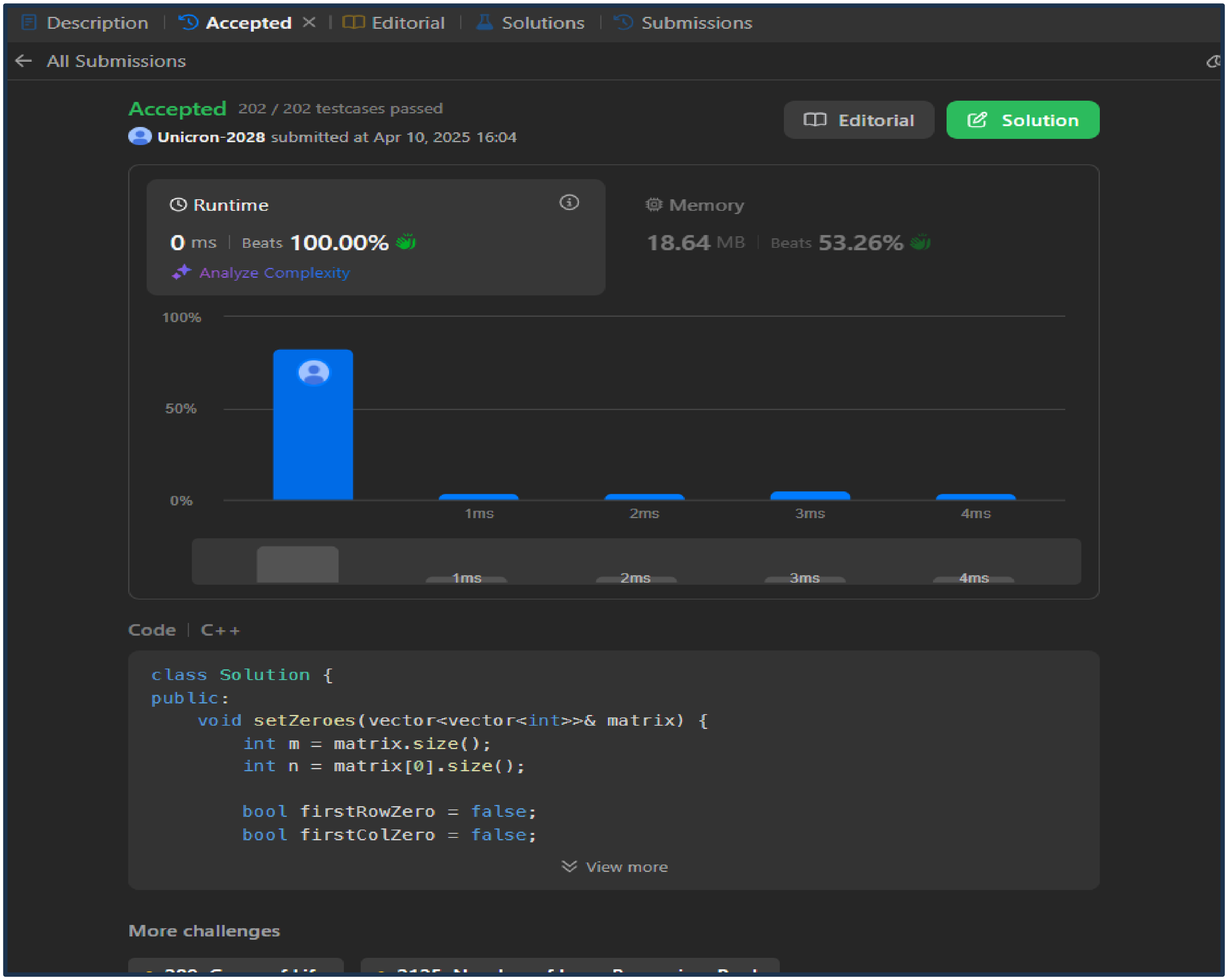
```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    bool hasCycle(ListNode *head) {
         if (!head || !head->next) return false;

        ListNode* slow = head;
        ListNode* fast = head;

        while (fast && fast->next) {
            slow = slow->next;
            fast = fast->next->next;

            if (slow == fast) {
                return true;
            }
        }

        return false;

    }
};
```

D. Longest substring without Repeating characters

```
class Solution {
public:
int lengthOfLongestSubstring(string s) {
unordered_map<char, int> lastSeen;
int maxLen = 0;
int start = 0;

for (int end = 0; end < s.length(); ++end) {
char currentChar = s[end];

if (lastSeen.find(currentChar) != lastSeen.end() &&
    lastSeen[currentChar] >= start) {
start = lastSeen[currentChar] + 1;
}

lastSeen[currentChar] = end;
maxLen = max(maxLen, end - start + 1);
}

return maxLen;
}
    };
```

3. Output

A. Set Matrix Zero



Accepted 202 / 202 testcases passed

Unicron-2028 submitted at Apr 10, 2025 16:04

Runtime: 0 ms | Beats 100.00%

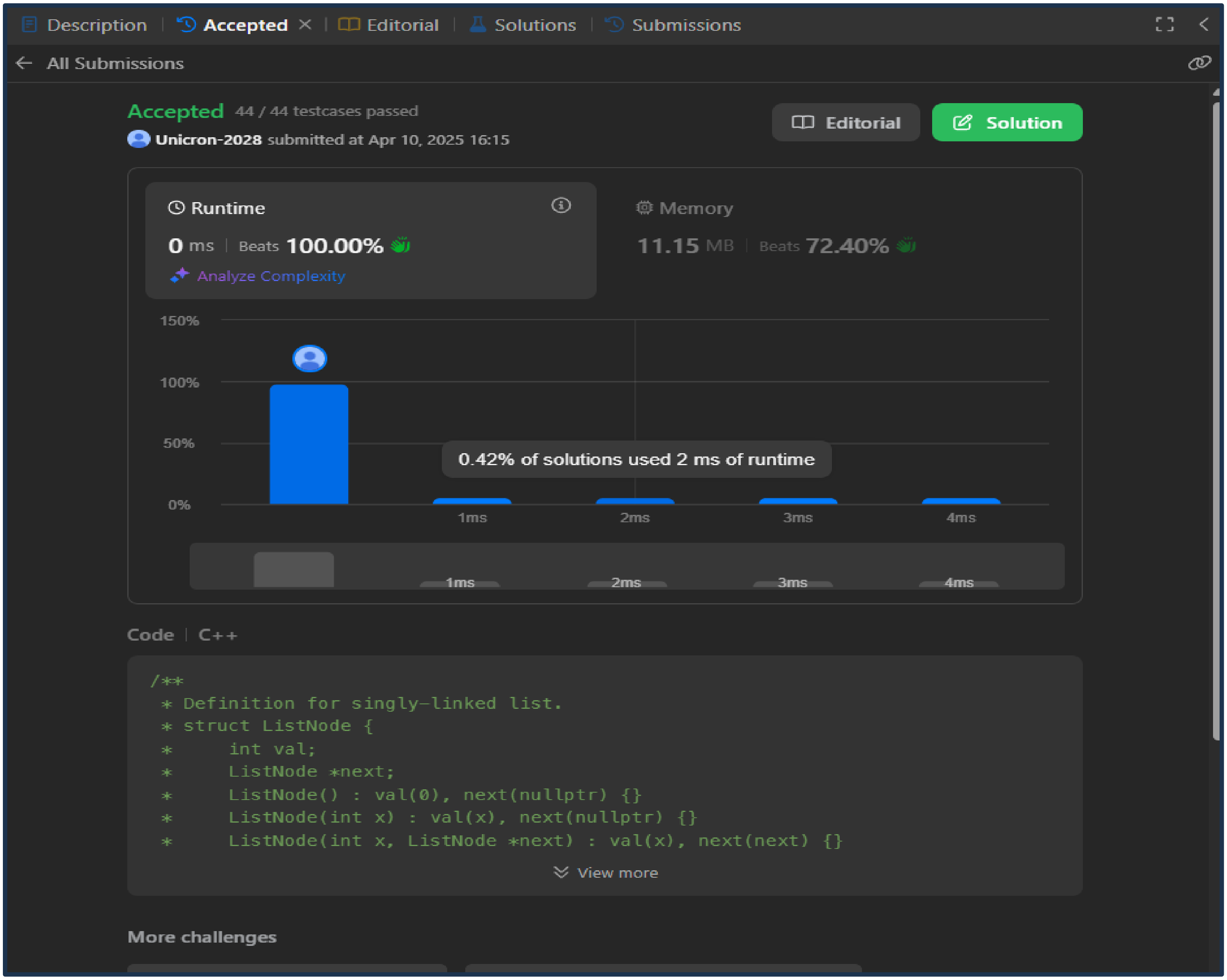
Memory: 18.64 MB | Beats 53.26%

Code | C++

```cpp
class Solution {
public:
    void setZeroes(vector<vector<int>>& matrix) {
        int m = matrix.size();
        int n = matrix[0].size();

        bool firstRowZero = false;
        bool firstColZero = false;
```

More challenges

B. Reverse Linked List II



Accepted 44 / 44 testcases passed

Unicron-2028 submitted at Apr 10, 2025 16:15

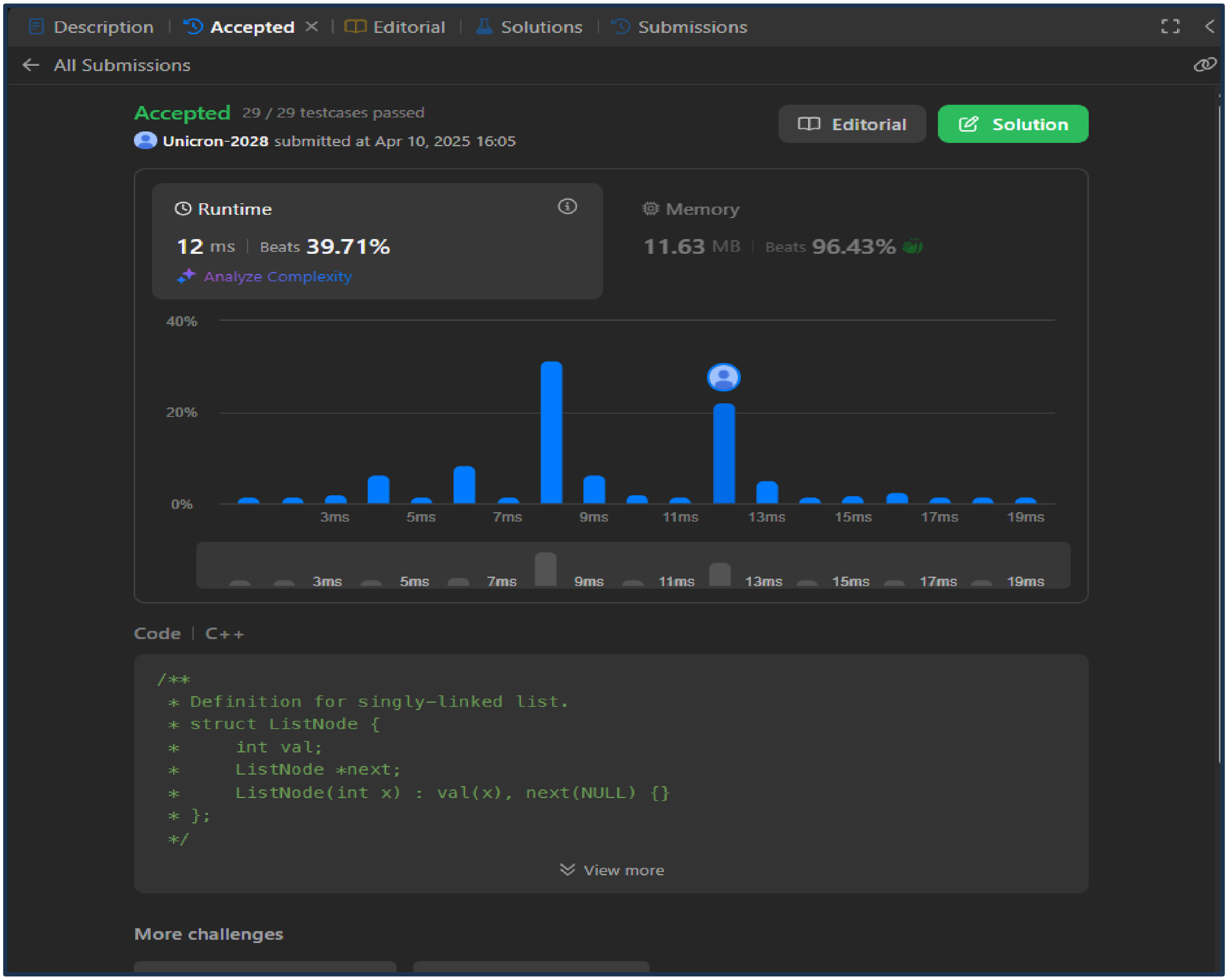
0 ms | Beats 100.00%

11.15 MB | Beats 72.40%

0.42% of solutions used 2 ms of runtime

```
/**
 * 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) {}
```

C. Linked List Cycle



Accepted 29 / 29 testcases passed

Unicron-2028 submitted at Apr 10, 2025 16:05

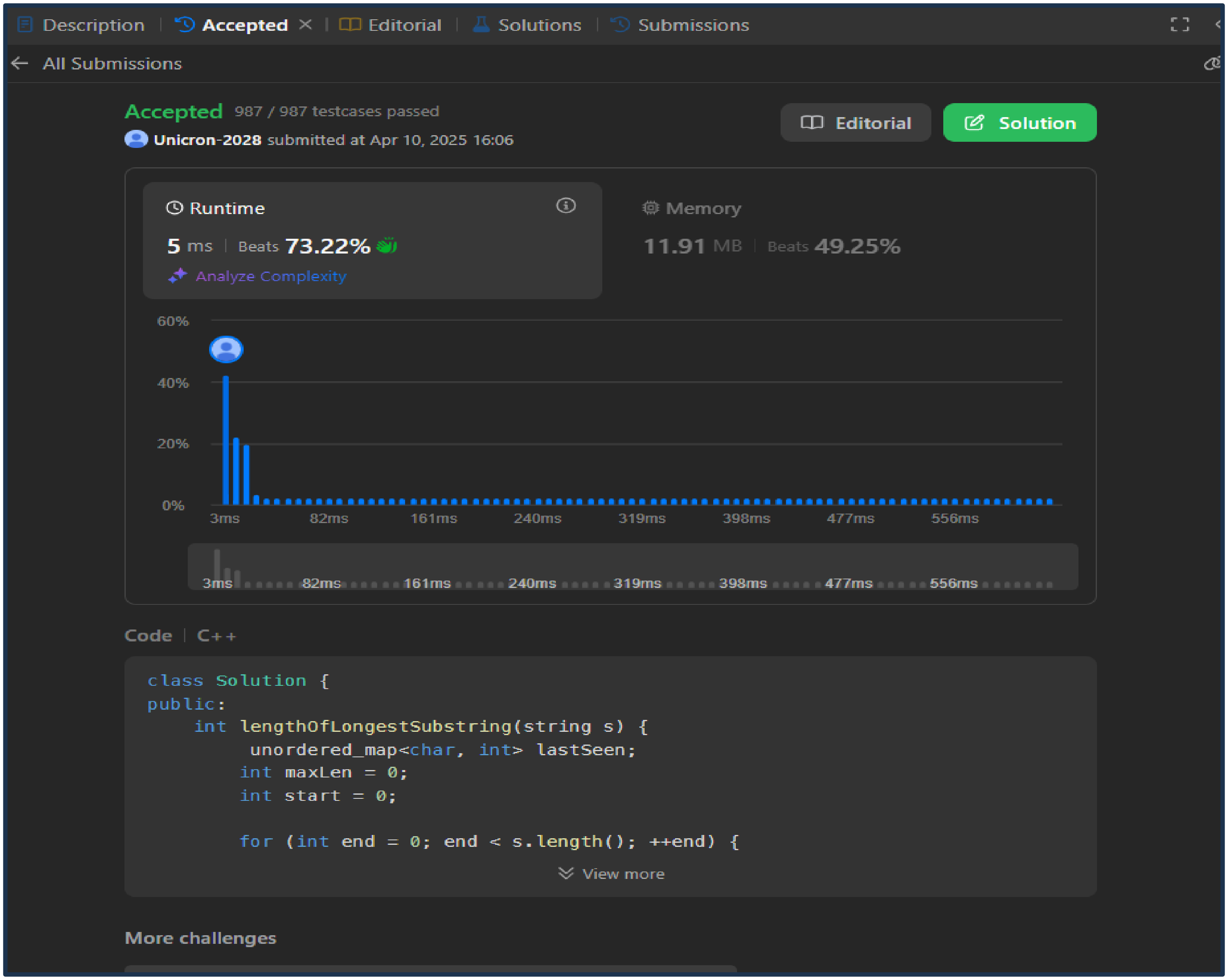
Runtime: 12 ms | Beats 39.71%

Memory: 11.63 MB | Beats 96.43%

Code | C++

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

D. Longest Substring without Repeating Characters



Accepted 987 / 987 testcases passed

Unicron-2028 submitted at Apr 10, 2025 16:06

Runtime

5 ms | Beats 73.22%

Memory

11.91 MB | Beats 49.25%

Code | C++

```
class Solution {
public:
    int lengthOfLongestSubstring(string s) {
         unordered_map<char, int> lastSeen;
        int maxLen = 0;
        int start = 0;

        for (int end = 0; end < s.length(); ++end) {
```

More challenges