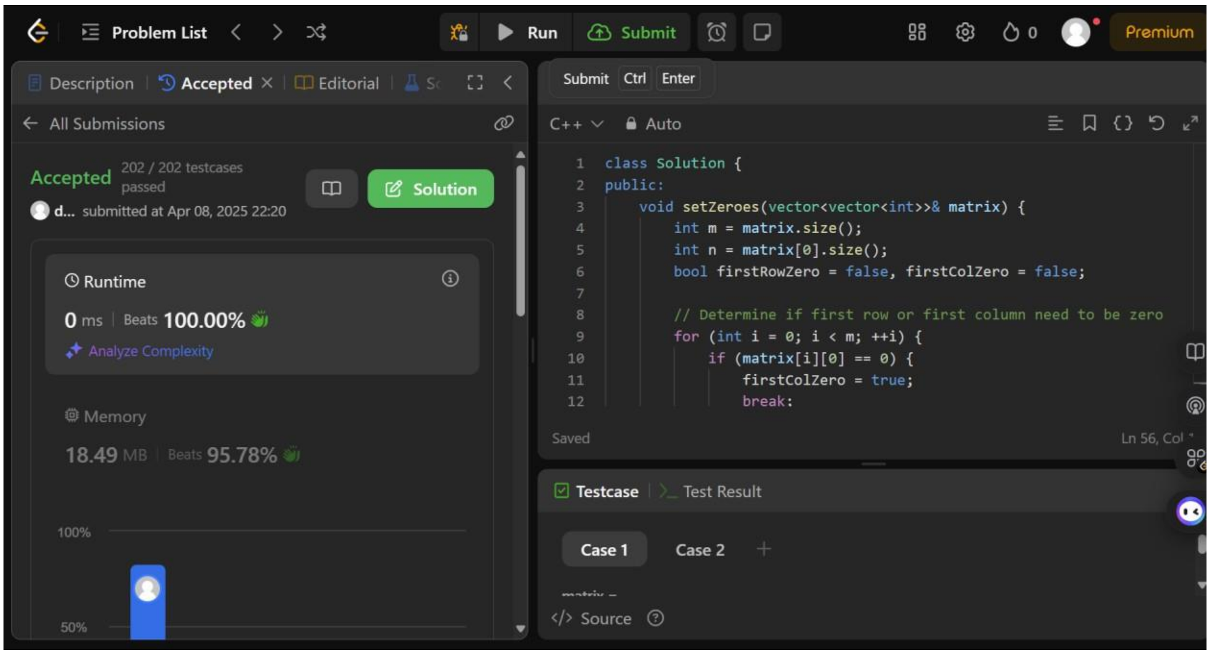
Hard Problems for Fast Learner

Name –Deepanshu Negi **UID-22BCS16773**

1. Set Matrix Zeroes: Given an m x n matrix, if an element is 0, set its entire row and column to 0.



Accepted 202 / 202 testcases passed

d... submitted at Apr 08, 2025 22:20

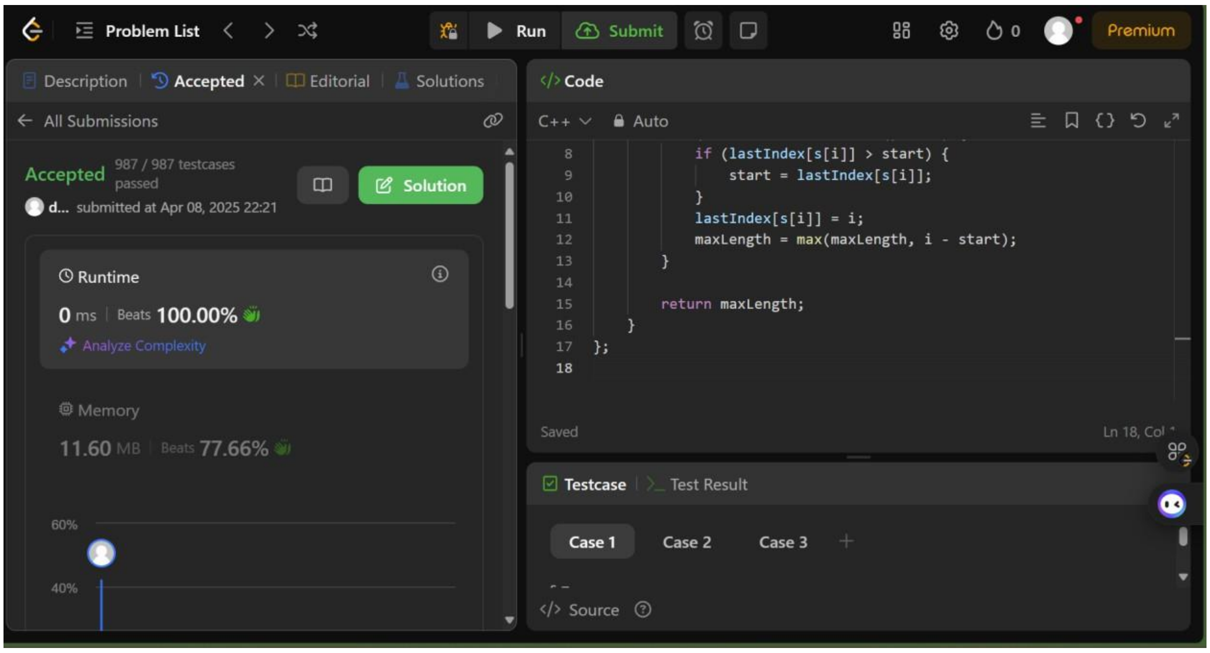
Runtime 0 ms | Beats 100.00%

Memory 18.49 MB | Beats 95.78%

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

        // Determine if first row or first column need to be zero
        for (int i = 0; i < m; ++i) {
            if (matrix[i][0] == 0) {
                firstColZero = true;
                break:
```

2. Longest Substring Without Repeating Characters: Given a string s, find the length of the longest substring that does not contain any repeating characters.



Accepted 987 / 987 testcases passed
d... submitted at Apr 08, 2025 22:21

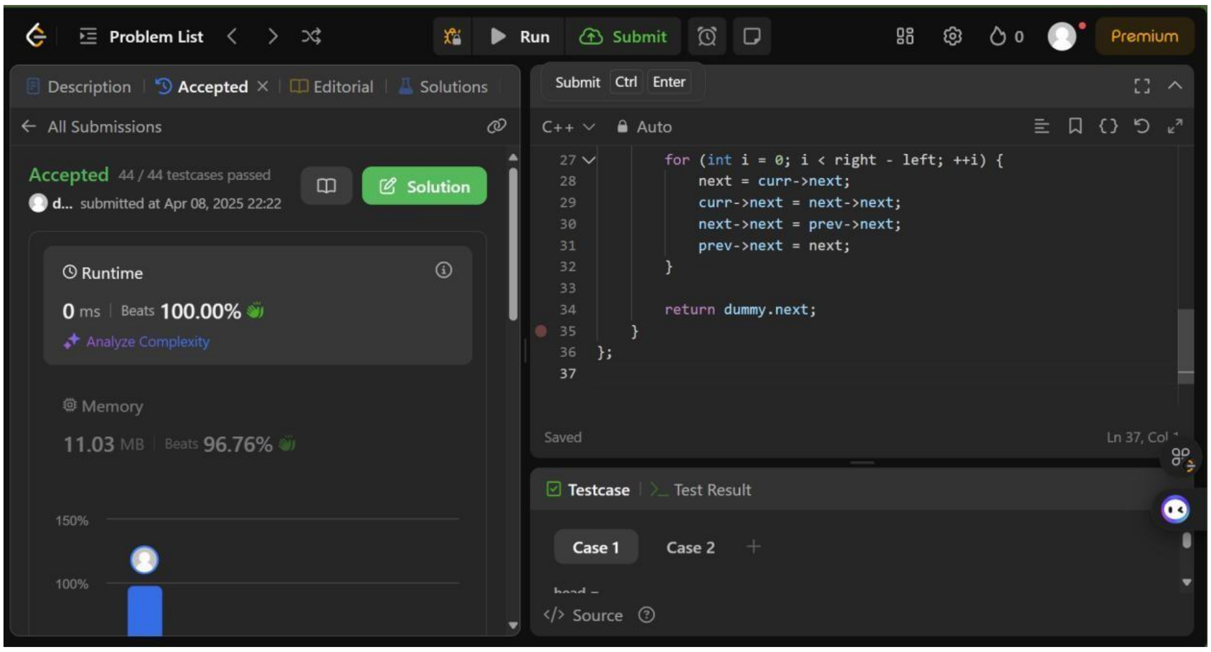
Runtime
0 ms | Beats 100.00%

Memory
11.60 MB | Beats 77.66%

```
            if (lastIndex[s[i]] > start) {
                start = lastIndex[s[i]];
            }
            lastIndex[s[i]] = i;
            maxLength = max(maxLength, i - start);
        }

        return maxLength;
    }
};
```

3. Reverse Linked List II: Given the head of a singly linked list and two integers left and right, reverse the nodes of the list from position left to right.



Accepted 44 / 44 testcases passed
d... submitted at Apr 08, 2025 22:22

Runtime
0 ms | Beats 100.00%

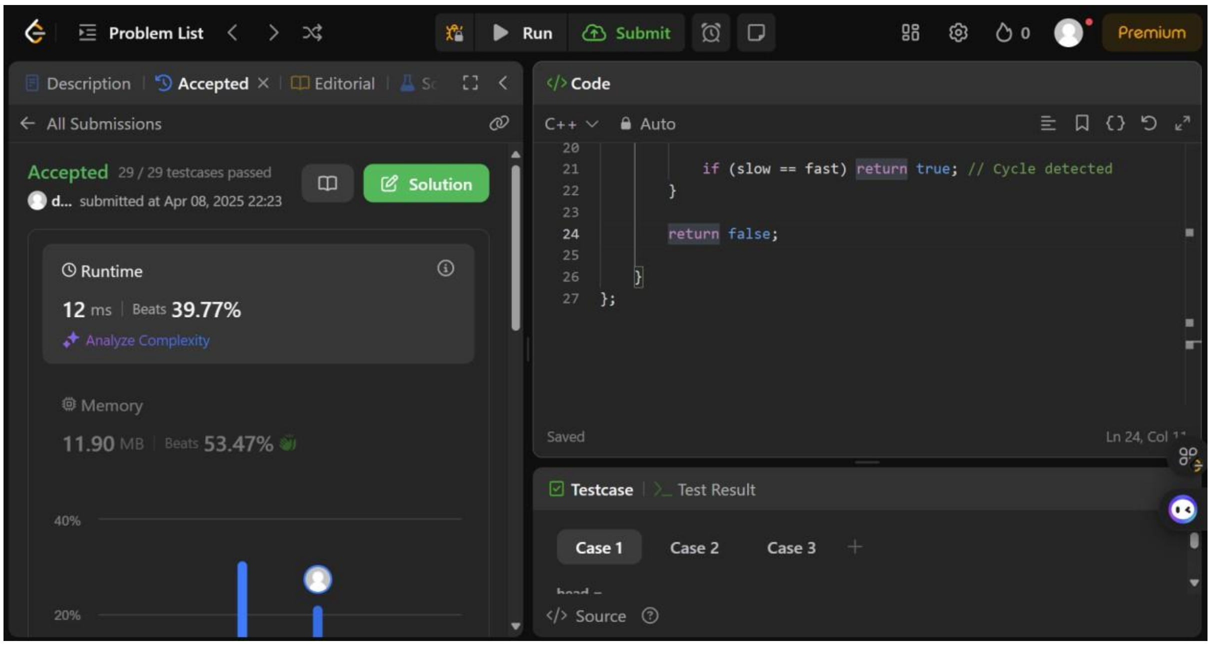
Memory
11.03 MB | Beats 96.76%

```
        for (int i = 0; i < right - left; ++i) {
            next = curr->next;
            curr->next = next->next;
            next->next = prev->next;
            prev->next = next;
        }

        return dummy.next;
    }
};
```

4. Detect a Cycle in a Linked List: Given the head of a linked list, determine whether the linked list contains a cycle. A cycle

occurs if a node's next pointer points to a previous node in the list.


Accepted 29 / 29 testcases passed

d... submitted at Apr 08, 2025 22:23

Runtime 12 ms | Beats 39.77%

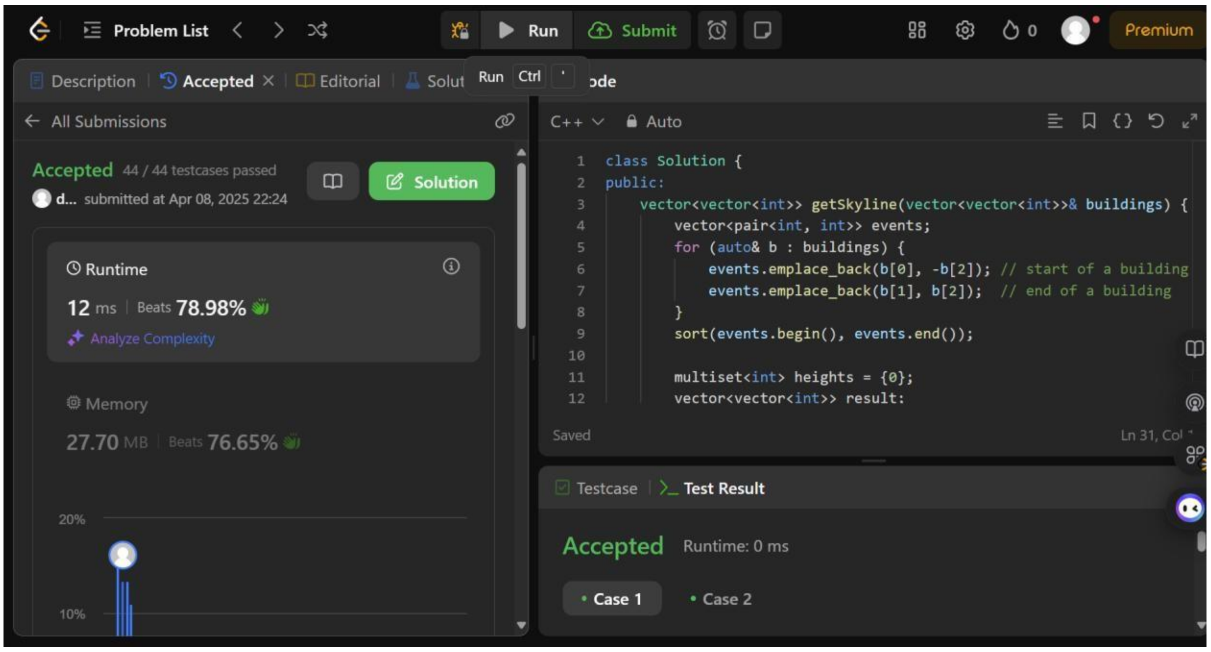
Memory 11.90 MB | Beats 53.47%

```
            if (slow == fast) return true; // Cycle detected
        }

        return false;
    }
};
```

Saved | Ln 24, Col 1

5. The Skyline Problem: Given a list of buildings represented as [left, right, height], where each building is a rectangle, return the key points of the skyline. A key point is represented as [x, y], where x is the x coordinate where the height changes to y


Accepted 44 / 44 testcases passed

d... submitted at Apr 08, 2025 22:24

Runtime 12 ms | Beats 78.98%

Memory 27.70 MB | Beats 76.65%

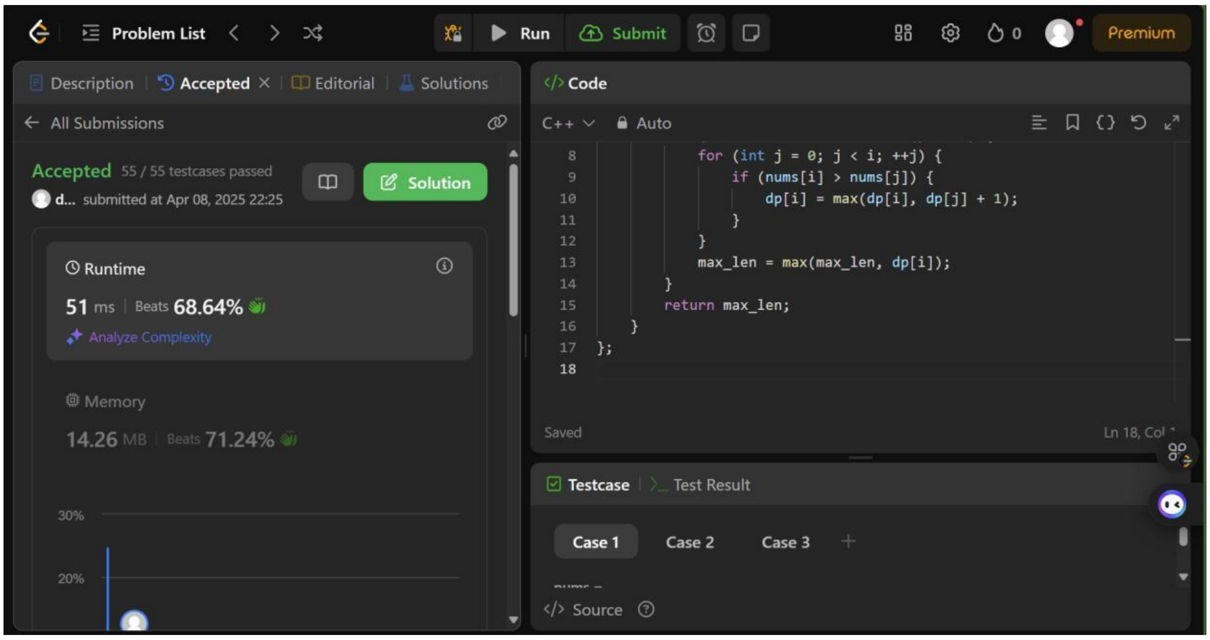
```
class Solution {
public:
    vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
        vector<pair<int, int>> events;
        for (auto& b : buildings) {
            events.emplace_back(b[0], -b[2]); // start of a building
            events.emplace_back(b[1], b[2]);  // end of a building
        }
        sort(events.begin(), events.end());

        multiset<int> heights = {0};
        vector<vector<int>> result:
```

Saved | Ln 31, Col

Accepted Runtime: 0 ms

6. Longest Increasing Subsequence II: Given an integer array nums, find the length of the longest strictly increasing subsequence. A subsequence is derived from the array by deleting some or no elements without changing the order of the remaining elements.



Accepted 55 / 55 testcases passed
d... submitted at Apr 08, 2025 22:25

Runtime
51 ms | Beats 68.64%

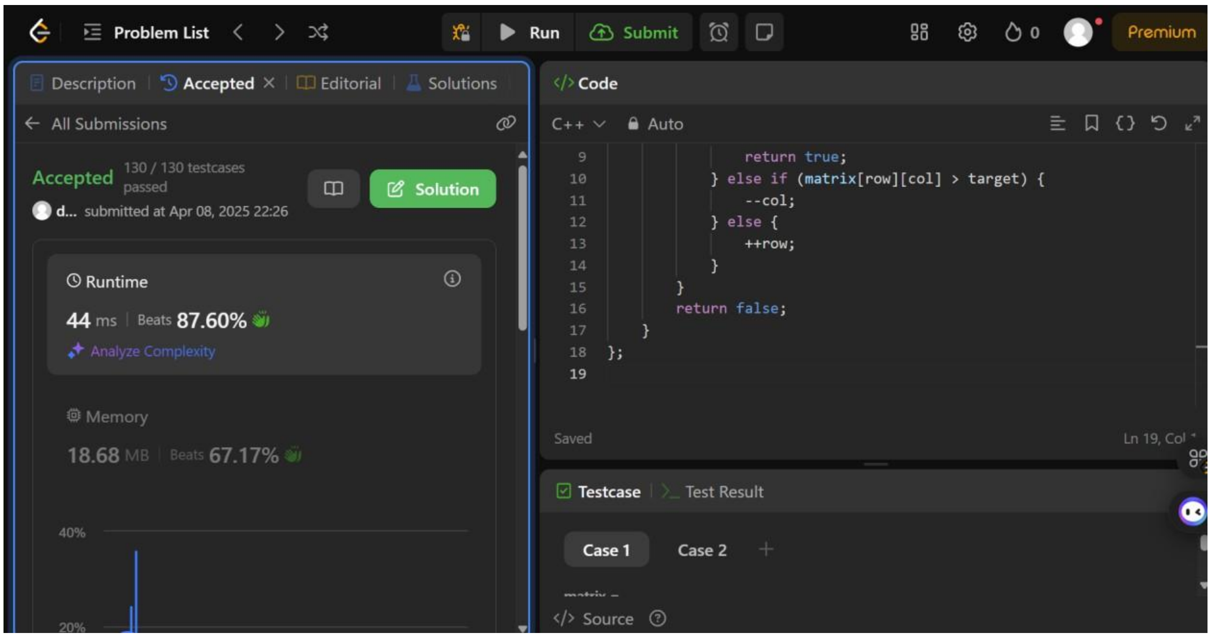
Memory
14.26 MB | Beats 71.24%

C++ Auto

```
            for (int j = 0; j < i; ++j) {
                if (nums[i] > nums[j]) {
                    dp[i] = max(dp[i], dp[j] + 1);
                }
            }
            max_len = max(max_len, dp[i]);
        }
        return max_len;
    }
};
```

Saved
Ln 18, Col 1

7. Search a 2D Matrix II: Given an m x n matrix where each row is sorted in ascending order from left to right and each column is sorted in ascending order from top to bottom, and an integer target, determine if the target exists in the matrix.



Accepted 130 / 130 testcases passed

d... submitted at Apr 08, 2025 22:26

Runtime

44 ms | Beats 87.60%

Analyze Complexity

Memory

18.68 MB | Beats 67.17%

40%

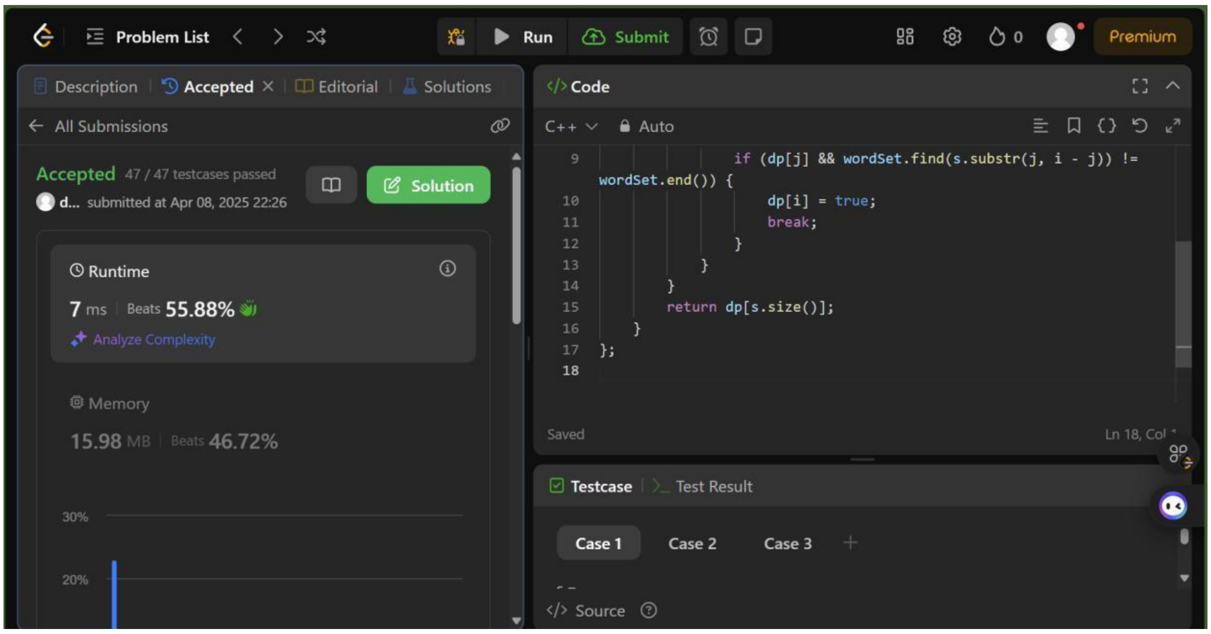
20%

```
                return true;
            } else if (matrix[row][col] > target) {
                --col;
            } else {
                ++row;
            }
        }
        return false;
    }
};
```

Saved

Ln 19, Col

7. Word Break: Given a string s and a dictionary wordDict containing a list of words, determine if s can be segmented into a space-separated sequence of one or more dictionary words. The same word can be reused multiple times.



Accepted 47 / 47 testcases passed

d... submitted at Apr 08, 2025 22:26

Runtime

7 ms | Beats 55.88%

Analyze Complexity

Memory

15.98 MB | Beats 46.72%

30%

20%

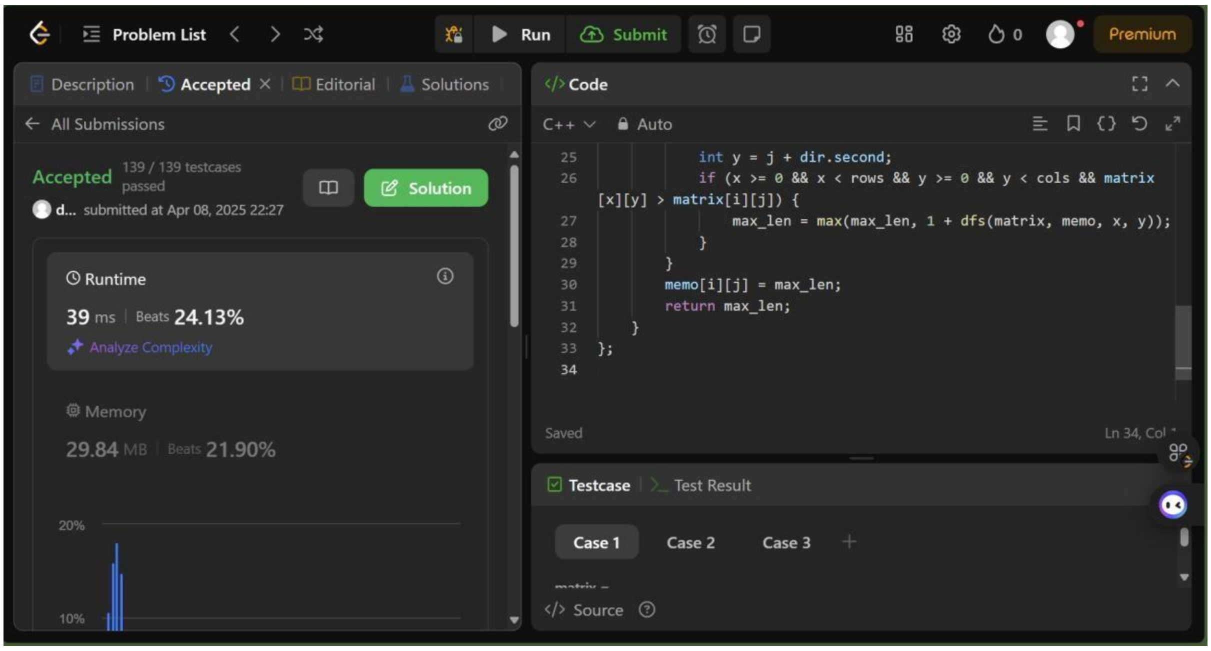
```
                if (dp[j] && wordSet.find(s.substr(j, i - j)) !=
wordSet.end()) {
                    dp[i] = true;
                    break;
                }
            }
        }
        return dp[s.size()];
    }
};
```

Saved

Ln 18, Col

8. Longest Increasing Path in a Matrix: Given an m x n integer matrix, find the length of the longest strictly increasing path. You can move up, down, left, or right from each cell. Diagonal

moves and moves outside the boundaries are not allowed.



Accepted 139 / 139 testcases passed

d... submitted at Apr 08, 2025 22:27

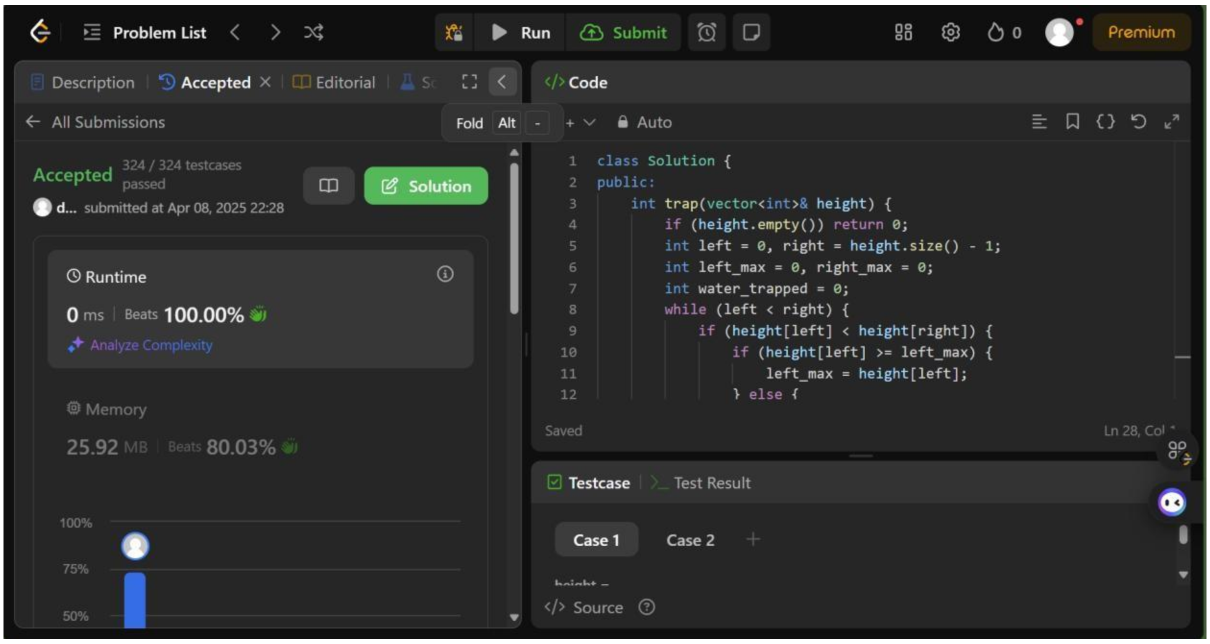
Runtime 39 ms | Beats 24.13%

Memory 29.84 MB | Beats 21.90%

```cpp
                int y = j + dir.second;
                if (x >= 0 && x < rows && y >= 0 && y < cols && matrix
[x][y] > matrix[i][j]) {
                    max_len = max(max_len, 1 + dfs(matrix, memo, x, y));
                }
            }
            memo[i][j] = max_len;
            return max_len;
        }
};

```

10. Trapping Rain Water: Given n non-negative integers representing an elevation map where the width of each bar is 1, compute the total amount of water that can be trapped after raining.



Accepted 324 / 324 testcases passed

d... submitted at Apr 08, 2025 22:28

Runtime 0 ms | Beats 100.00%

Memory 25.92 MB | Beats 80.03%

```cpp
class Solution {
public:
    int trap(vector<int>& height) {
        if (height.empty()) return 0;
        int left = 0, right = height.size() - 1;
        int left_max = 0, right_max = 0;
        int water_trapped = 0;
        while (left < right) {
            if (height[left] < height[right]) {
                if (height[left] >= left_max) {
                    left_max = height[left];
                } else {
```