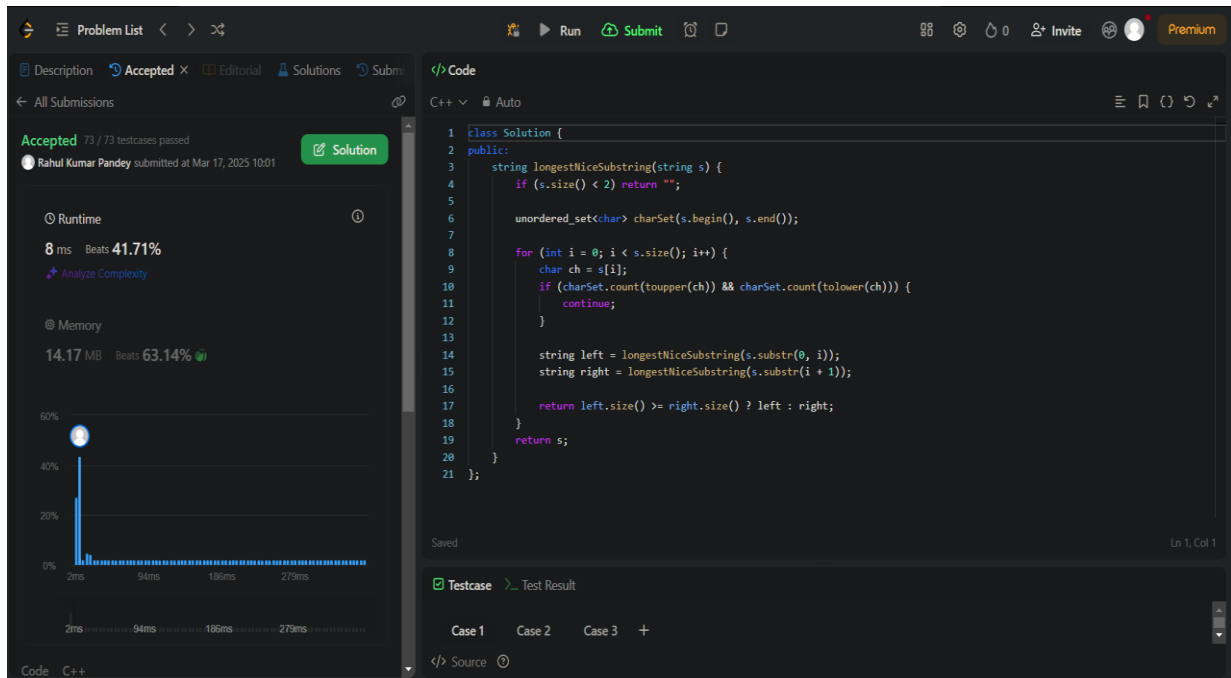
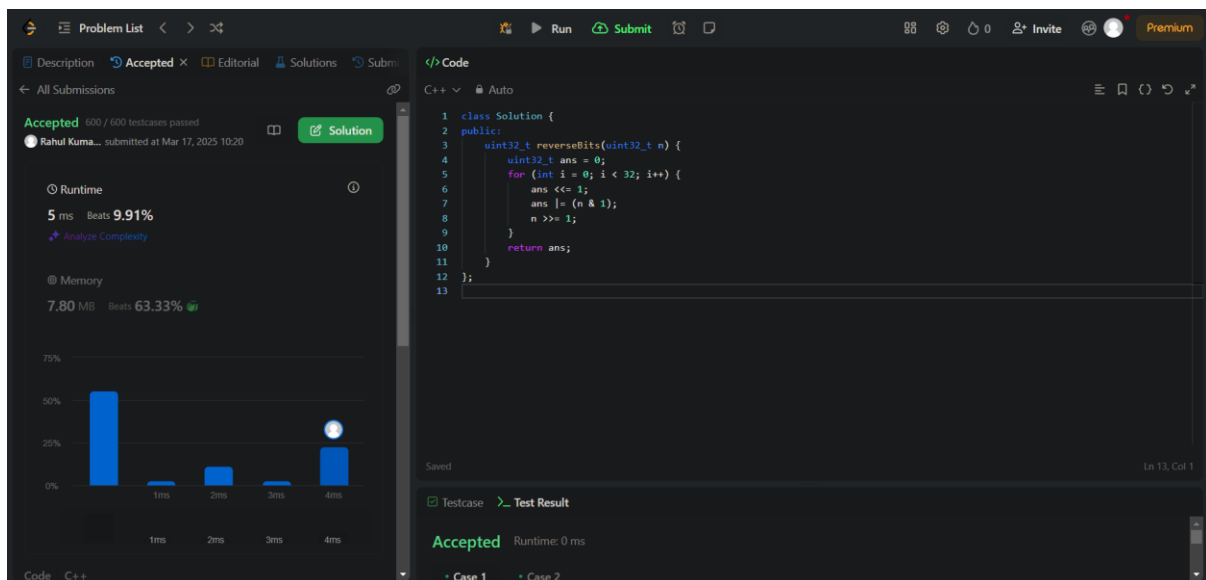


ASSIGNMENT 4

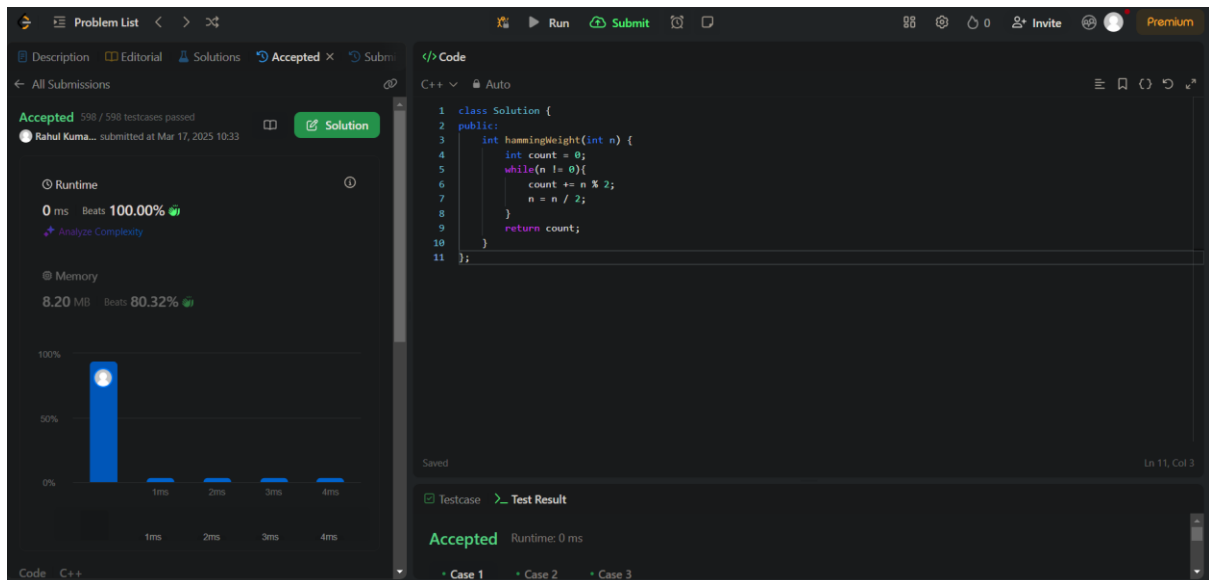
1) Longest Nice Substring



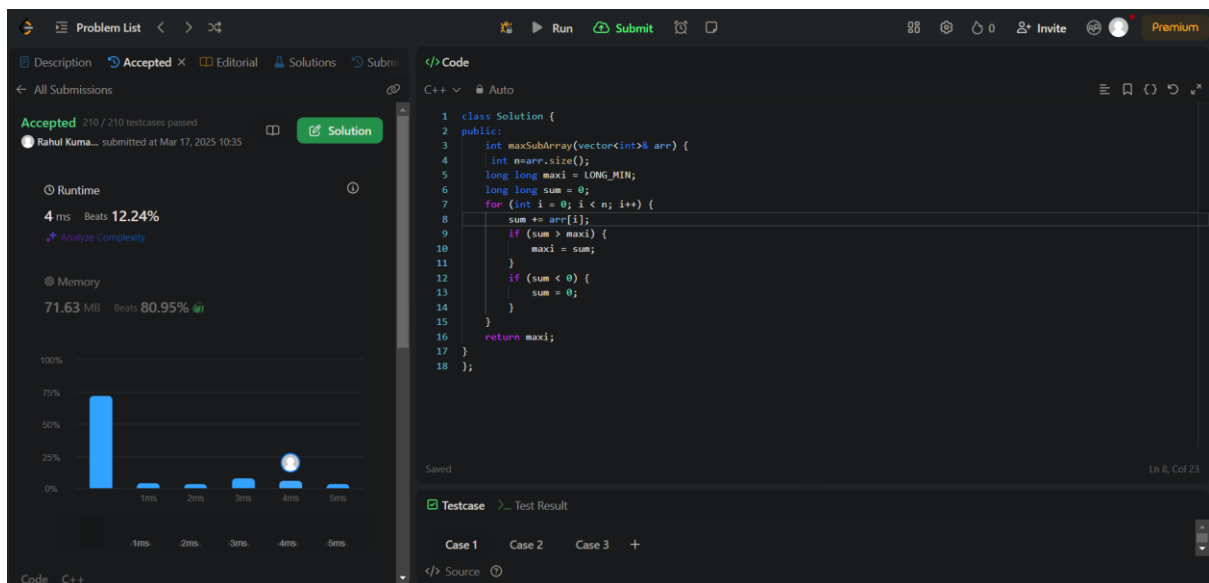
2) Reverse Bits



3) Number of 1 Bits



4) Maximum Subarray



5) Search a 2D Matrix II

The screenshot shows a C++ solution for the problem "Search a 2D Matrix II". The code is as follows:

```
1 class Solution {
2 public:
3     bool searchMatrix(vector<vector<int>>& matrix, int target) {
4
5         int n = matrix.size();
6         int m = matrix[0].size();
7         int row = 0, col = m - 1;
8         while (row < n && col >= 0) {
9             if (matrix[row][col] == target) return true;
10            else if (matrix[row][col] < target) row++;
11            else col--;
12        }
13        return false;
14    }
15};
```

Performance metrics: Runtime 57 ms, Beats 38.77%; Memory 18.73 MB, Beats 36.77%.

A bar chart shows the distribution of runtime times, with a peak at 4ms.

6) Super Pow

The screenshot shows a C++ solution for the problem "Super Pow". The code is as follows:

```
1 class Solution {
2 public:
3     int find(int a, int b)
4     {
5         a%=1337;
6         int res=1;
7         for(int i=0;i<b;i++)
8         {
9             res*=a;
10            res%=1337;
11        }
12        return res;
13    }
14    int superPow(int a, vector<int>& b) {
15        int res=1,x,f;
16        for(int i=0;i<b.size();i++)
17        {
18            x=find(a,b[i]);
19            x*=res;
20            x%=1337;
21            f=x;
22            x=find(x,10);
23            res=x;
24        }
25        return f;
26    }
27};
```

Performance metrics: Runtime 0 ms, Beats 100.00%; Memory 15.15 MB, Beats 83.54%.

A bar chart shows the distribution of runtime times, with a peak at 0ms.

7) Beautiful Array

Problem List < > ✕

Description Editorial Solutions Accepted × Submissions

← All Submissions

Accepted 38 / 38 testcases passed

● Rahul Kuma... submitted at Mar 17, 2025 13:14

Runtime 11 ms Beats 19.37%
✚ Analyze Complexity

Memory 16.62 MB Beats 18.37%

75%
50%
25%
0%

0ms 5ms 10ms 15ms

Code C++

```
1 class Solution {
2 public:
3     vector<int> beautifulArray(int n) {
4         if(n==1)
5             return {1};
6         vector<int> even = beautifulArray(n/2);
7         vector<int> odd = beautifulArray(n-(n/2));
8         vector<int> ans;
9         for(auto e:even)
10             ans.push_back(2*e);
11         for(auto o:odd)
12             ans.push_back((2*o)-1);
13         return ans;
14     }
15 }
```

Saved Ln 14, Col 6

Testcase > Test Result

Case 1 Case 2 +

</> Source ⓘ

8) The Skyline Problem

Problem List < > ✕

Description Editorial Solutions Accepted × Submissions

← All Submissions

Accepted 44 / 44 testcases passed

● Rahul Kuma... submitted at Mar 17, 2025 13:17

Runtime 23 ms Beats 41.24%
✚ Analyze Complexity

Memory 28.66 MB Beats 55.40% @j

20%
10%
0%

1ms 71ms 140ms 210ms

Code C++

```
1 class Solution {
2 public:
3     vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
4         vector<vector<int>> ans;
5         multiset<int> pq(0);
6
7         vector<pair<int, int>> points;
8
9         for(auto b: buildings){
10             points.push_back({b[0], -b[2]});
11             points.push_back({b[1], b[2]});
12         }
13
14         sort(points.begin(), points.end());
15
16         int ongoingHeight = 0;
17
18         for(int i = 0; i < points.size(); i++){
19             int currentPoint = points[i].first;
20             int heightAtCurrentPoint = points[i].second;
21
22             if(heightAtCurrentPoint < 0){
23                 pq.insert(-heightAtCurrentPoint);
24             } else {
25                 pq.erase(pq.find(heightAtCurrentPoint));
26             }
27
28             auto next = *pq.begin();
29         }
```

Saved Ln 28, Col 1

Testcase > Test Result

10)Reverse Pairs

The screenshot shows a LeetCode solution for the 'Reverse Pairs' problem. The solution is implemented in C++ using a merge sort algorithm to count reverse pairs. The left sidebar displays the submission status as 'Accepted' with 140/140 testcases passed, submitted by 'Rahul Kuma...' on Mar 17, 2025. Performance metrics show a runtime of 227 ms (beats 83.43%) and memory usage of 113.51 MB (beats 79.37%). A runtime histogram is also visible. The main editor shows the C++ code, which includes a 'mergeSort' function to count reverse pairs and a 'merge' function to merge two sorted arrays. The bottom status bar indicates the code is saved and ready for testing.

Accepted 140 / 140 testcases passed
Rahul Kuma... submitted at Mar 17, 2025 13:19

Runtime
227 ms Beats 83.43%
[Analyze Complexity](#)

Memory
113.51 MB Beats 79.37%

6%
4%
2%
0%
3ms 156ms 313ms 468ms

```
1 class Solution {
2 public:
3     int reversePairs(vector<int>& nums) {
4         return mergeSort(nums, 0, nums.size() - 1);
5     }
6 private:
7     int mergeSort(vector<int>& nums, int low, int high) {
8         int cnt = 0;
9         if (low >= high) return cnt;
10        int mid = low + (high - low) / 2;
11        cnt += mergeSort(nums, low, mid); // left half
12        cnt += mergeSort(nums, mid + 1, high); // right half
13        cnt += countPairs(nums, low, mid, high);
14        merge(nums, low, mid, high);
15        return cnt;
16    }
17
18    void merge(vector<int>& nums, int low, int mid, int high) {
19        vector<int> temp(high - low + 1);
20        int left = low;
21        int right = mid + 1;
22        int index = 0;
23
24        while (left <= mid && right <= high) {
25            if (nums[left] <= nums[right]) {
26                temp[index++] = nums[left++];
27            } else {
28                temp[index++] = nums[right++];
29            }
30        }
31        while (left <= mid) temp[index++] = nums[left++];
32        while (right <= high) temp[index++] = nums[right++];
33        for (int i = low; i <= high; i++) nums[i] = temp[i - low];
34    }
35
36    int countPairs(vector<int>& nums, int low, int mid, int high) {
37        int cnt = 0;
38        int j = mid + 1;
39        for (int i = low; i <= mid; i++) {
40            while (j <= high && nums[i] > 2 * nums[j]) j++;
41            cnt += j - mid - 1;
42        }
43        return cnt;
44    }
45 }
```

Saved Ln 12, Col 6

☒ Testcase [Test Result](#)