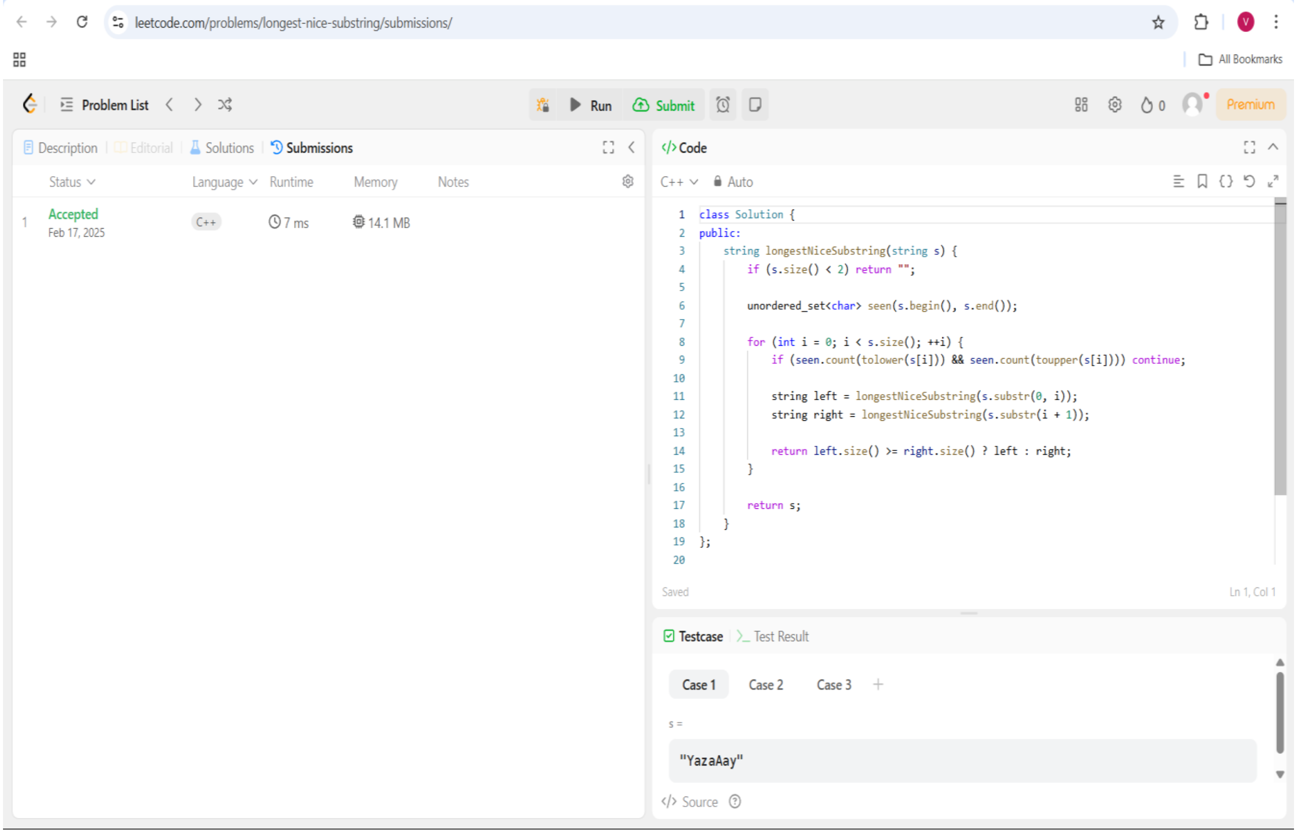
DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
CU CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

Assignment-4

Name – Vedant Aggarwal **UID –** 22BCS13945
Semester - 6 **Date -** 17-03-2025
Subject - Advanced Programming Lab-2 **Subject Code -** 22CSP-351

- **Divide and Conquer:**

1. Longest Nice Substring-



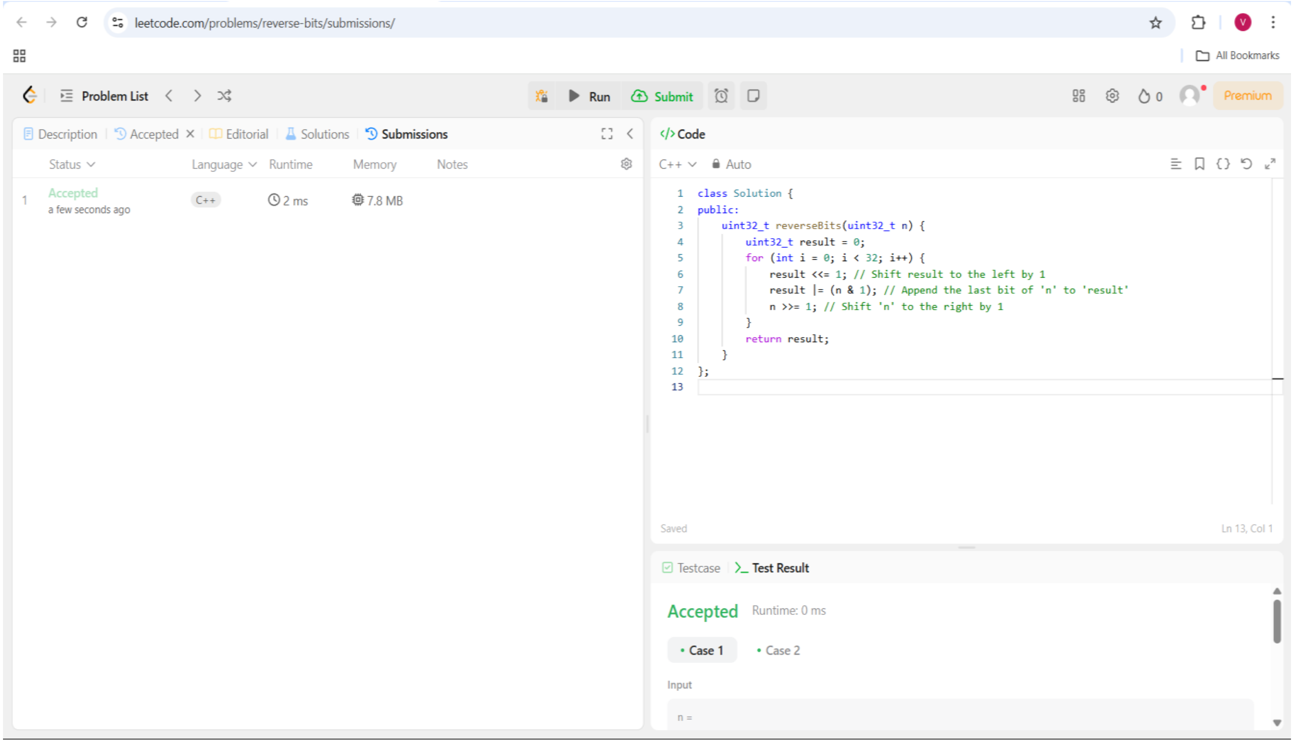
	Status	Language	Runtime	Memory	Notes
1	Accepted Feb 17, 2025	C++	7 ms	14.1 MB	

```cpp
class Solution {
public:
    string longestNiceSubstring(string s) {
        if (s.size() < 2) return "";

        unordered_set<char> seen(s.begin(), s.end());

        for (int i = 0; i < s.size(); ++i) {
            if (seen.count(tolower(s[i])) && seen.count(toupper(s[i]))) continue;

            string left = longestNiceSubstring(s.substr(0, i));
            string right = longestNiceSubstring(s.substr(i + 1));

            return left.size() >= right.size() ? left : right;
        }

        return s;
    }
};
```

s =

"YazaAay"

2. Reverse Bits-



	Status	Language	Runtime	Memory	Notes
1	Accepted a few seconds ago	C++	2 ms	7.8 MB	

```cpp
class Solution {
public:
    uint32_t reverseBits(uint32_t n) {
        uint32_t result = 0;
        for (int i = 0; i < 32; i++) {
            result <<= 1; // Shift result to the left by 1
            result |= (n & 1); // Append the last bit of 'n' to 'result'
            n >>= 1; // Shift 'n' to the right by 1
        }
        return result;
    }
};
```

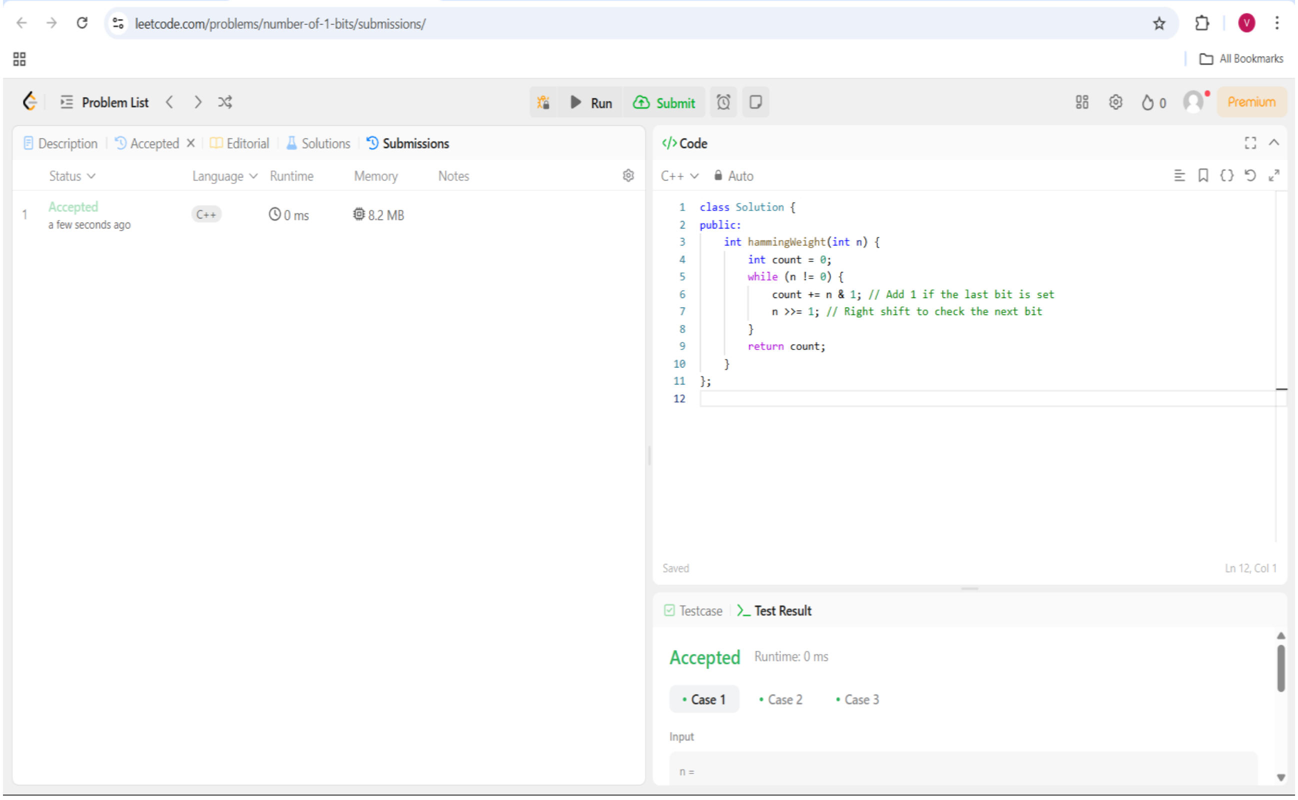
Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

n =

3. Number of 1 Bits-



	Status	Language	Runtime	Memory	Notes
1	Accepted a few seconds ago	C++	0 ms	8.2 MB	

```cpp
class Solution {
public:
    int hammingWeight(int n) {
        int count = 0;
        while (n != 0) {
            count += n & 1; // Add 1 if the last bit is set
            n >>= 1; // Right shift to check the next bit
        }
        return count;
    }
};
```

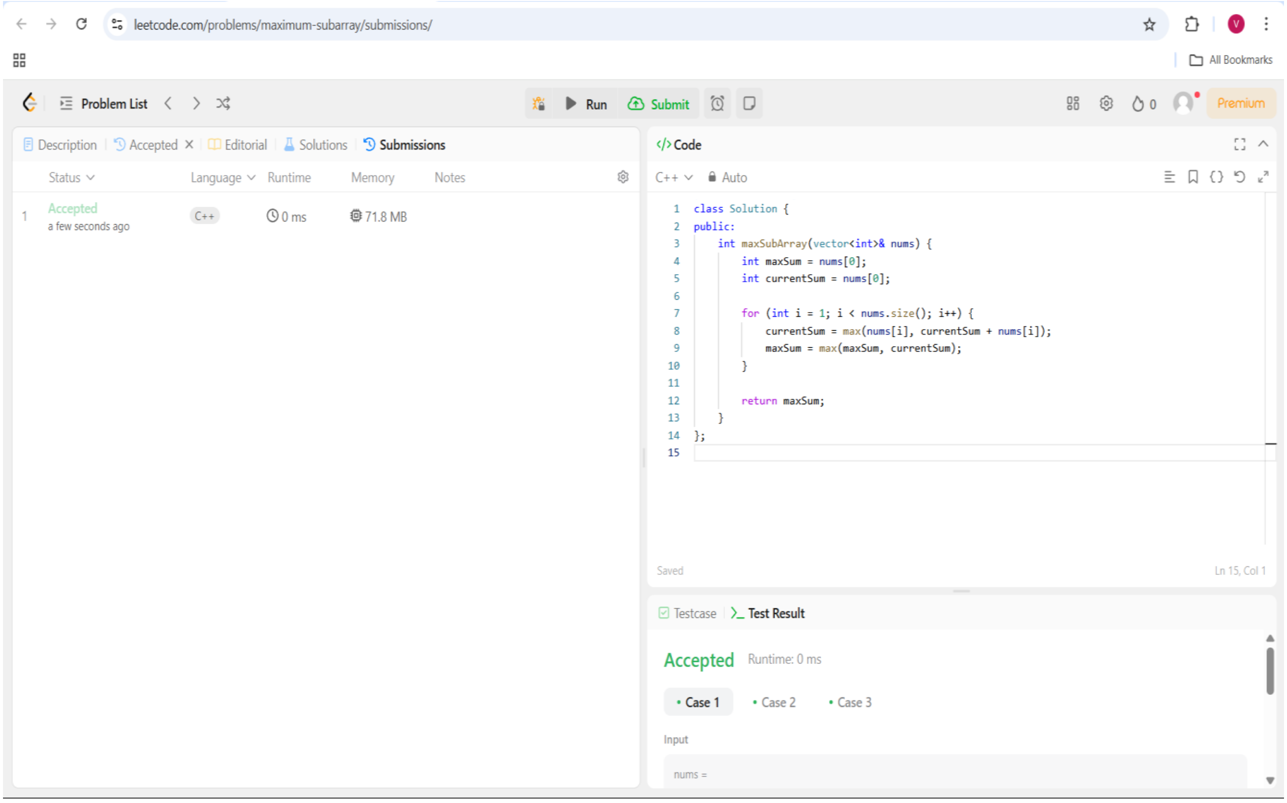
Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

n =

4. Maximum Subarray-



	Status	Language	Runtime	Memory	Notes
1	Accepted a few seconds ago	C++	0 ms	71.8 MB	

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int maxSum = nums[0];
        int currentSum = nums[0];

        for (int i = 1; i < nums.size(); i++) {
            currentSum = max(nums[i], currentSum + nums[i]);
            maxSum = max(maxSum, currentSum);
        }

        return maxSum;
    }
};
```

Saved Ln 15, Col 1

Testcase Test Result

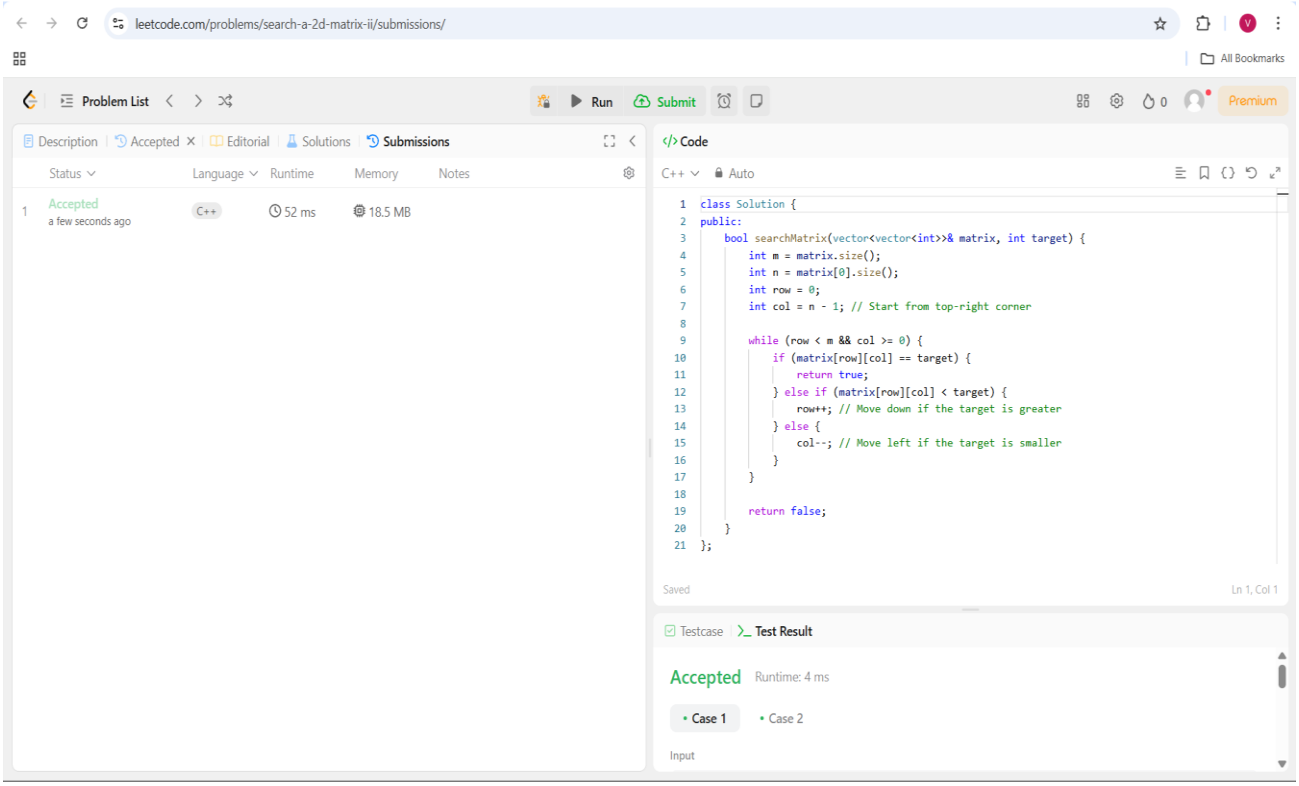
Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

nums =

5. Search a 2D Matrix II-



	Status	Language	Runtime	Memory	Notes
1	Accepted a few seconds ago	C++	52 ms	18.5 MB	

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int m = matrix.size();
        int n = matrix[0].size();
        int row = 0;
        int col = n - 1; // Start from top-right corner

        while (row < m && col >= 0) {
            if (matrix[row][col] == target) {
                return true;
            } else if (matrix[row][col] < target) {
                row++; // Move down if the target is greater
            } else {
                col--; // Move left if the target is smaller
            }
        }

        return false;
    }
};
```

Saved Ln 1, Col 1

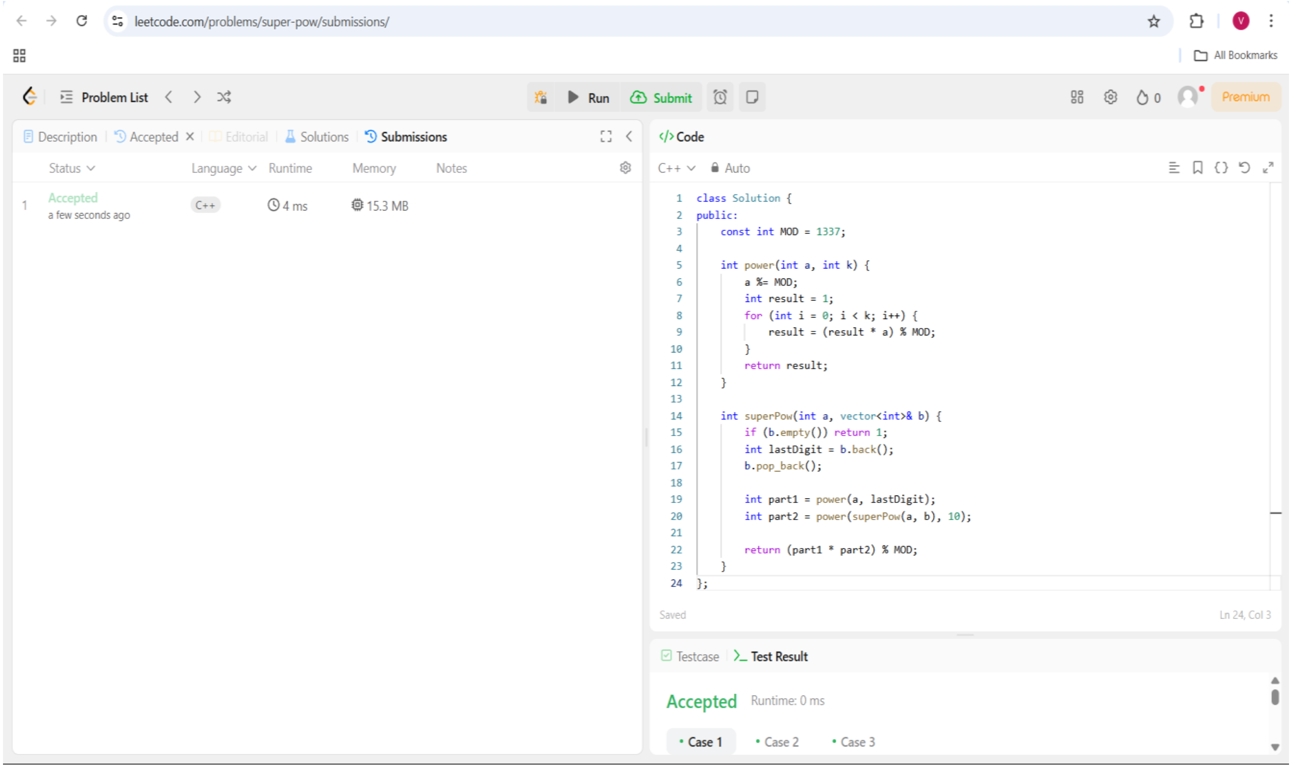
Testcase Test Result

Accepted Runtime: 4 ms

Case 1 Case 2

Input

6. Super Pow-



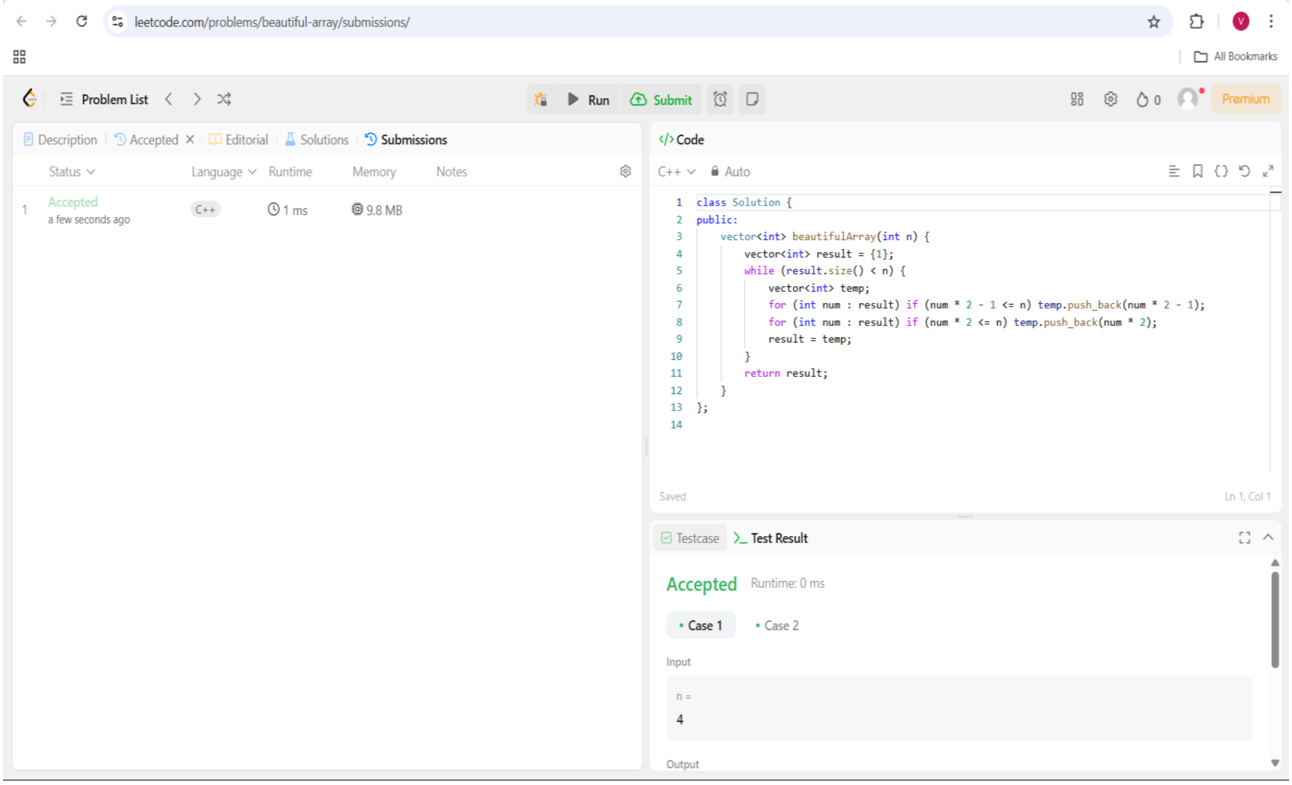
	Status	Language	Runtime	Memory	Notes
1	Accepted a few seconds ago	C++	4 ms	15.3 MB	

```
class Solution {
public:
    const int MOD = 1337;

    int power(int a, int k) {
        a %= MOD;
        int result = 1;
        for (int i = 0; i < k; i++) {
            result = (result * a) % MOD;
        }
        return result;
    }

    int superPow(int a, vector<int>& b) {
        if (b.empty()) return 1;
        int lastDigit = b.back();
        b.pop_back();

        int part1 = power(a, lastDigit);
        int part2 = power(superPow(a, b), 10);

        return (part1 * part2) % MOD;
    }
};
```

Saved Ln 24, Col 3

Accepted Runtime: 0 ms

7. Beautiful Array-



	Status	Language	Runtime	Memory	Notes
1	Accepted a few seconds ago	C++	1 ms	9.8 MB	

```
class Solution {
public:
    vector<int> beautifulArray(int n) {
        vector<int> result = {1};
        while (result.size() < n) {
            vector<int> temp;
            for (int num : result) if (num * 2 - 1 <= n) temp.push_back(num * 2 - 1);
            for (int num : result) if (num * 2 <= n) temp.push_back(num * 2);
            result = temp;
        }
        return result;
    }
};
```

Saved Ln 1, Col 1

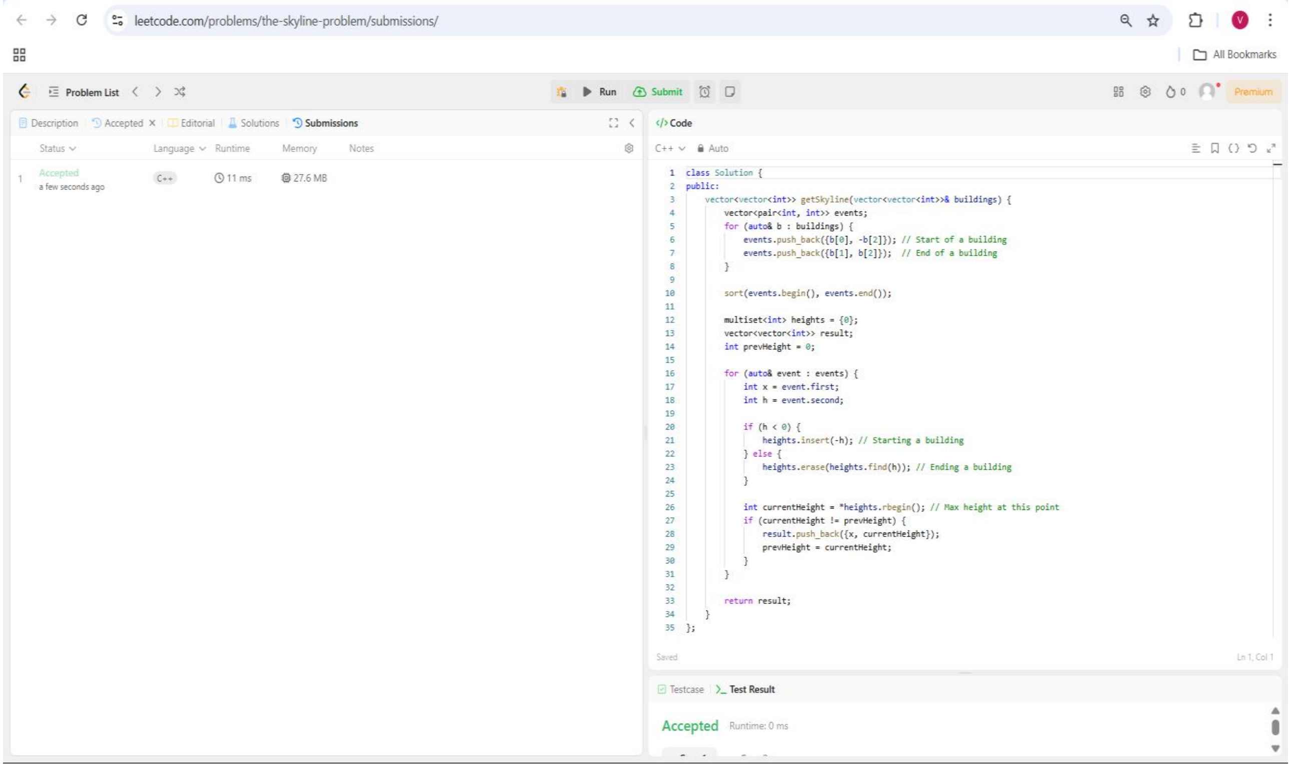
Accepted Runtime: 0 ms

Input

n =
4

Output

8. The Skyline Problem-

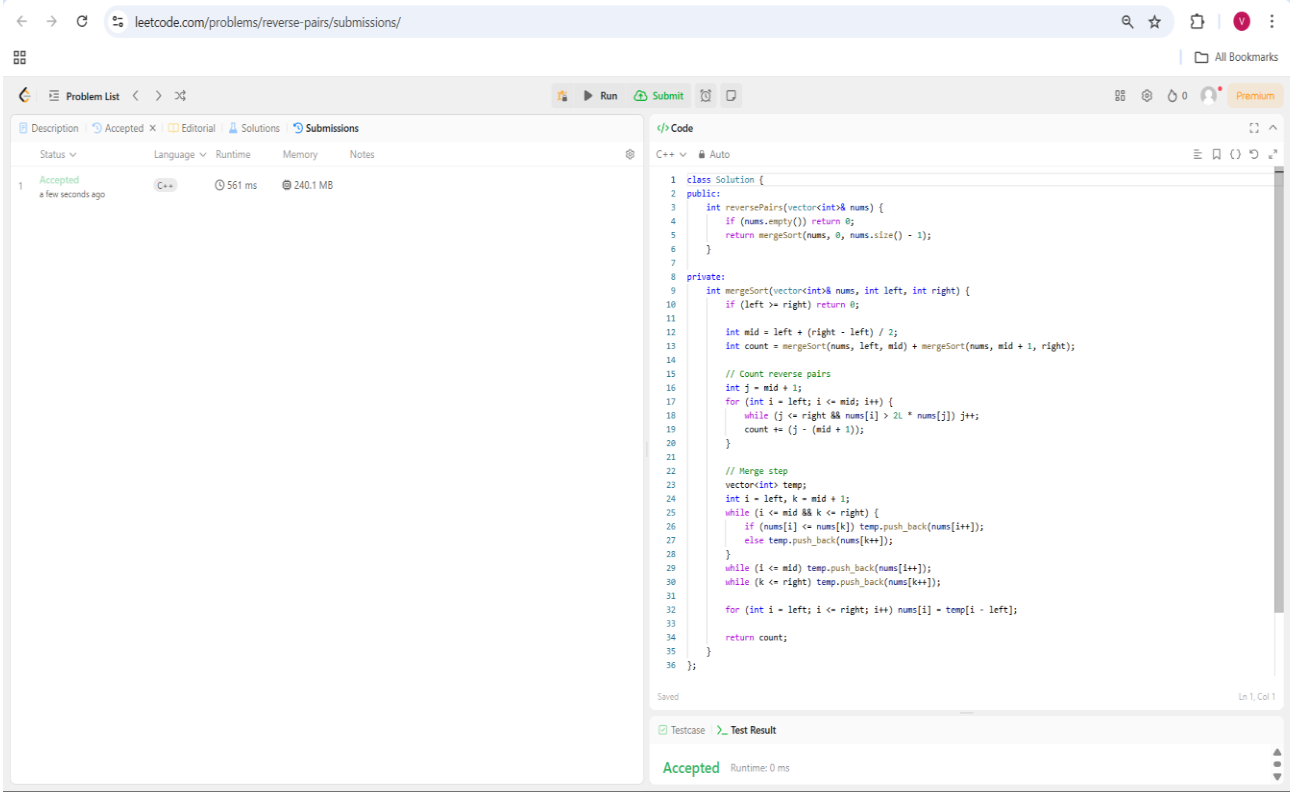


1 Accepted a few seconds ago C++ 11 ms 27.6 MB

```cpp
class Solution {
public:
    vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
        vector<pair<int, int>> events;
        for (auto& b : buildings) {
            events.push_back({b[0], -b[2]}); // Start of a building
            events.push_back({b[1], b[2]});  // End of a building
        }

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

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

        for (auto& event : events) {
            int x = event.first;
            int h = event.second;

            if (h < 0) {
                heights.insert(-h); // Starting a building
            } else {
                heights.erase(heights.find(h)); // Ending a building
            }

            int currentHeight = *heights.rbegin(); // Max height at this point
            if (currentHeight != prevHeight) {
                result.push_back({x, currentHeight});
                prevHeight = currentHeight;
            }
        }

        return result;
    }
};
```

Accepted Runtime: 0 ms

9. Reverse Pairs-

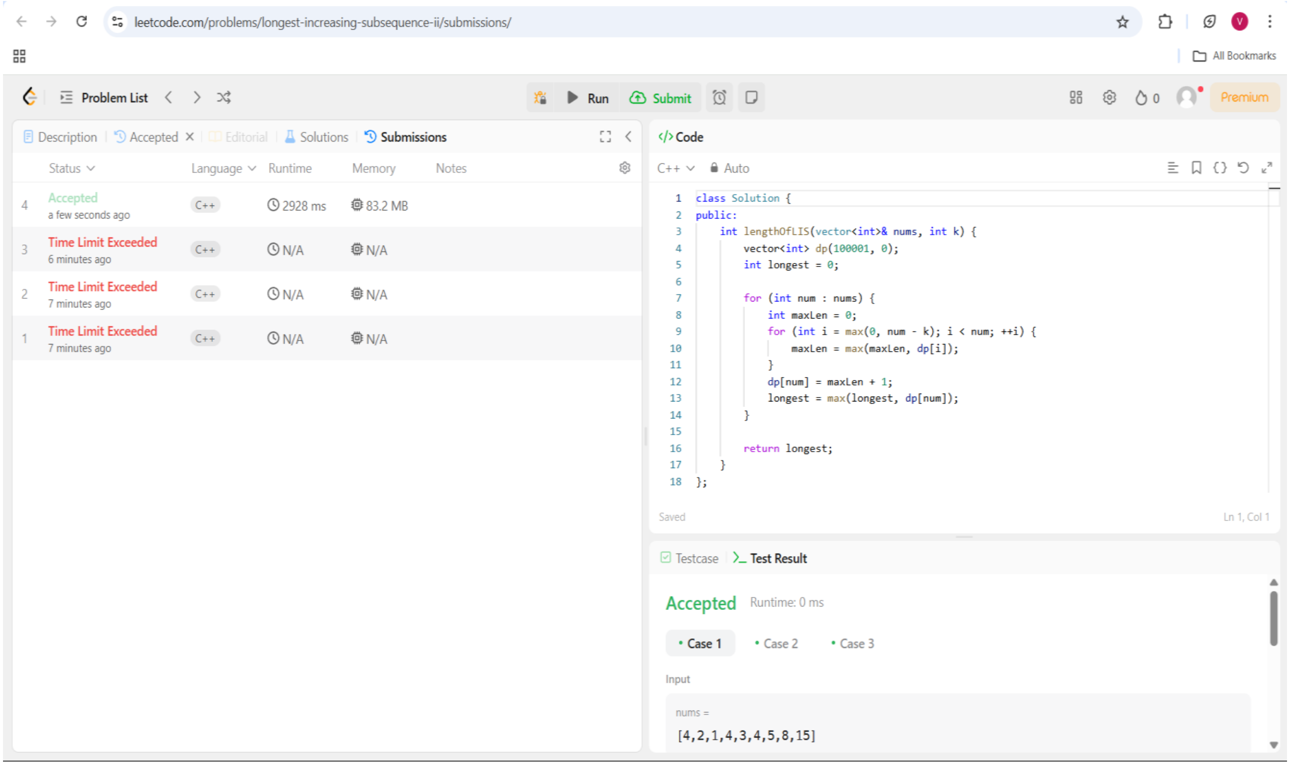


1 Accepted a few seconds ago C++ 561 ms 240.1 MB

```cpp
class Solution {
public:
    int reversePairs(vector<int>& nums) {
        if (nums.empty()) return 0;
        return mergeSort(nums, 0, nums.size() - 1);
    }

private:
    int mergeSort(vector<int>& nums, int left, int right) {
        if (left >= right) return 0;

        int mid = left + (right - left) / 2;
        int count = mergeSort(nums, left, mid) + mergeSort(nums, mid + 1, right);

        // Count reverse pairs
        int j = mid + 1;
        for (int i = left; i <= mid; i++) {
            while (j <= right && nums[i] > 2L * nums[j]) j++;
            count += (j - (mid + 1));
        }

        // Merge step
        vector<int> temp;
        int i = left, k = mid + 1;
        while (i <= mid && k <= right) {
            if (nums[i] <= nums[k]) temp.push_back(nums[i++]);
            else temp.push_back(nums[k++]);
        }
        while (i <= mid) temp.push_back(nums[i++]);
        while (k <= right) temp.push_back(nums[k++]);

        for (int i = left; i <= right; i++) nums[i] = temp[i - left];

        return count;
    }
};
```

Accepted Runtime: 0 ms

10. **Longest Increasing Subsequence II-**



leetcode.com/problems/longest-increasing-subsequence-ii/submissions/

	Status	Language	Runtime	Memory	Notes
4	Accepted a few seconds ago	C++	2928 ms	83.2 MB	
3	Time Limit Exceeded 6 minutes ago	C++	N/A	N/A	
2	Time Limit Exceeded 7 minutes ago	C++	N/A	N/A	
1	Time Limit Exceeded 7 minutes ago	C++	N/A	N/A	

```cpp
class Solution {
public:
    int lengthOfLIS(vector<int>& nums, int k) {
        vector<int> dp(100001, 0);
        int longest = 0;

        for (int num : nums) {
            int maxLen = 0;
            for (int i = max(0, num - k); i < num; ++i) {
                maxLen = max(maxLen, dp[i]);
            }
            dp[num] = maxLen + 1;
            longest = max(longest, dp[num]);
        }

        return longest;
    }
};
```

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

nums =

[4,2,1,4,3,4,5,8,15]