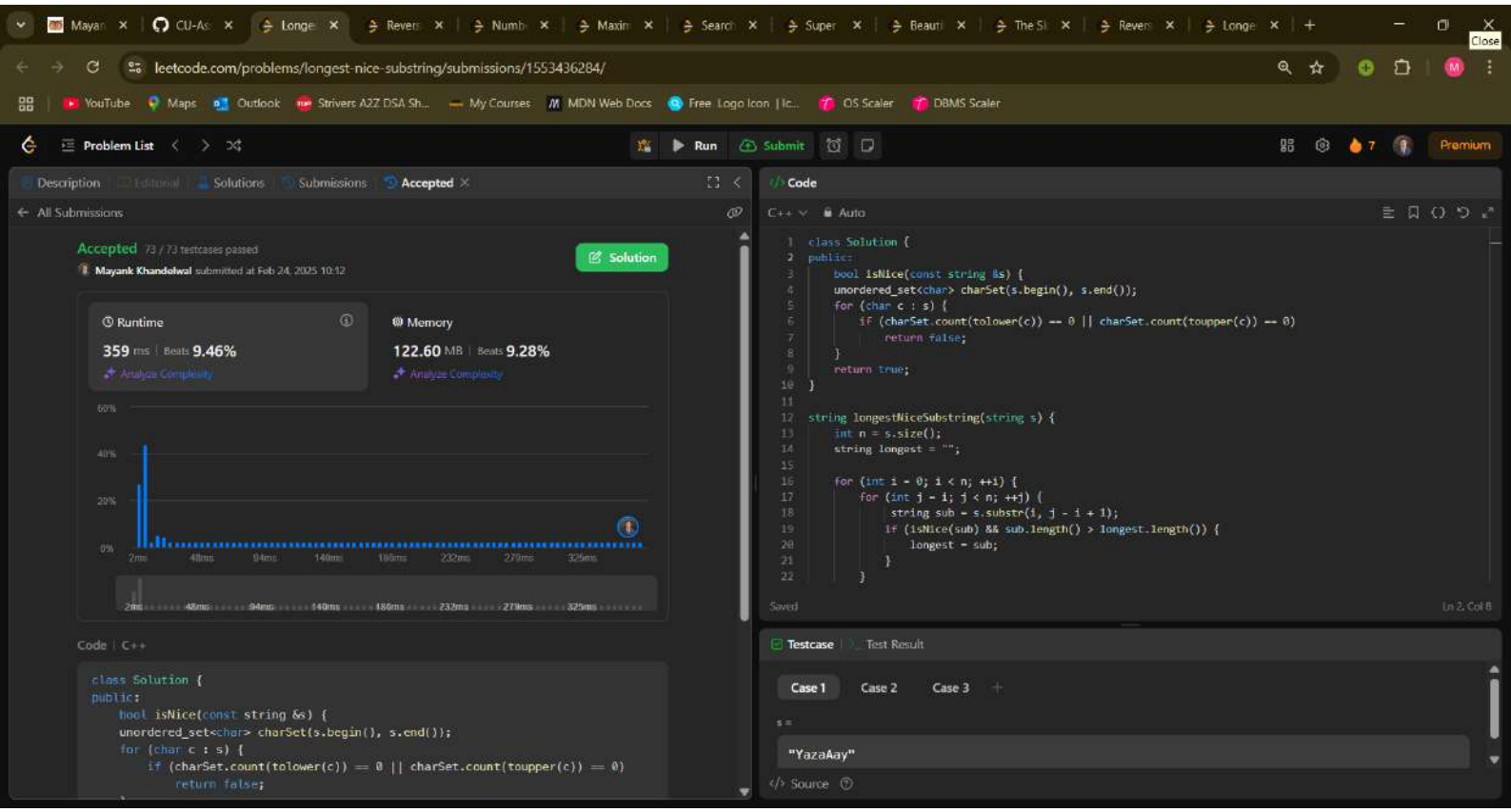
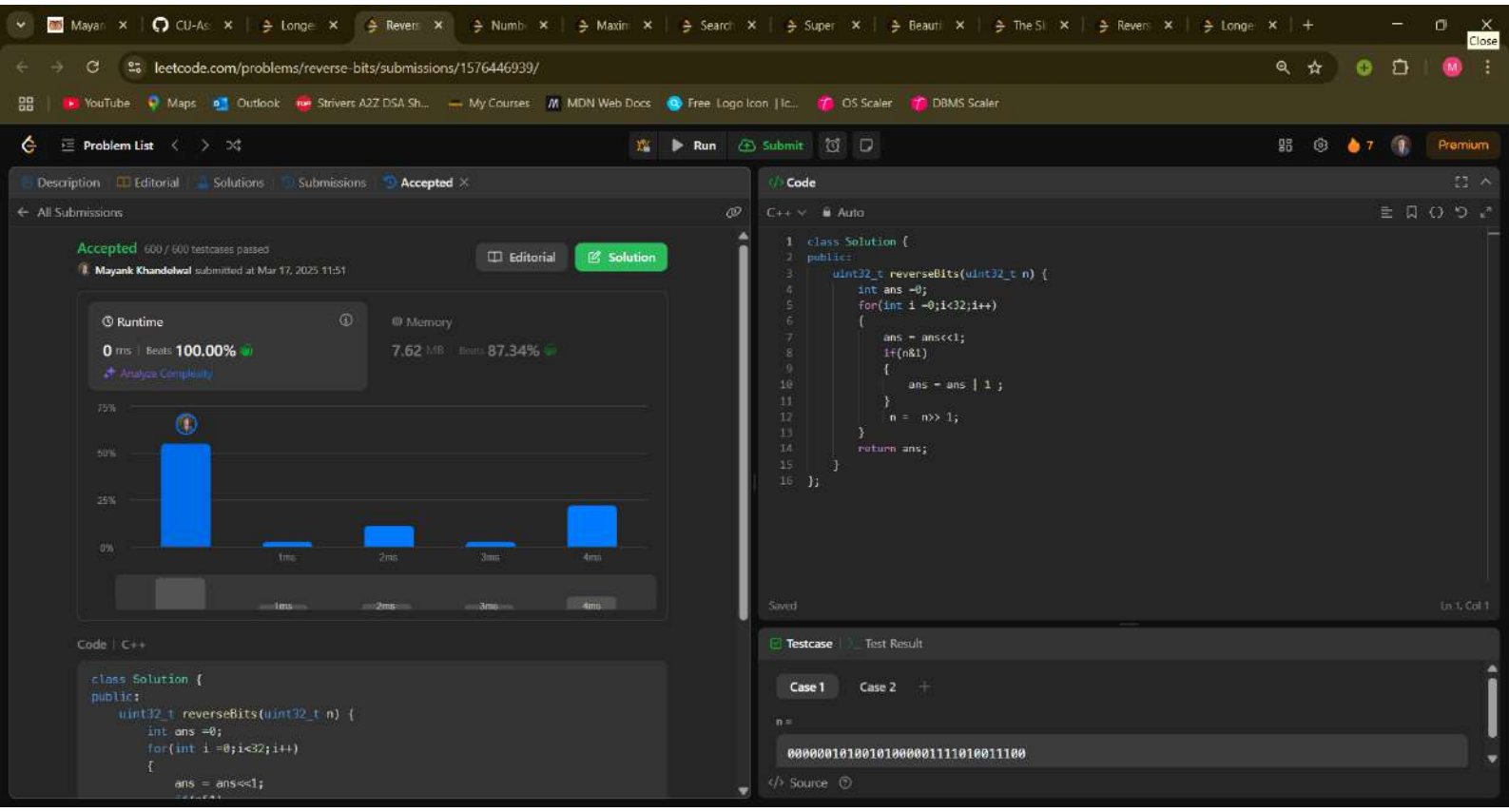
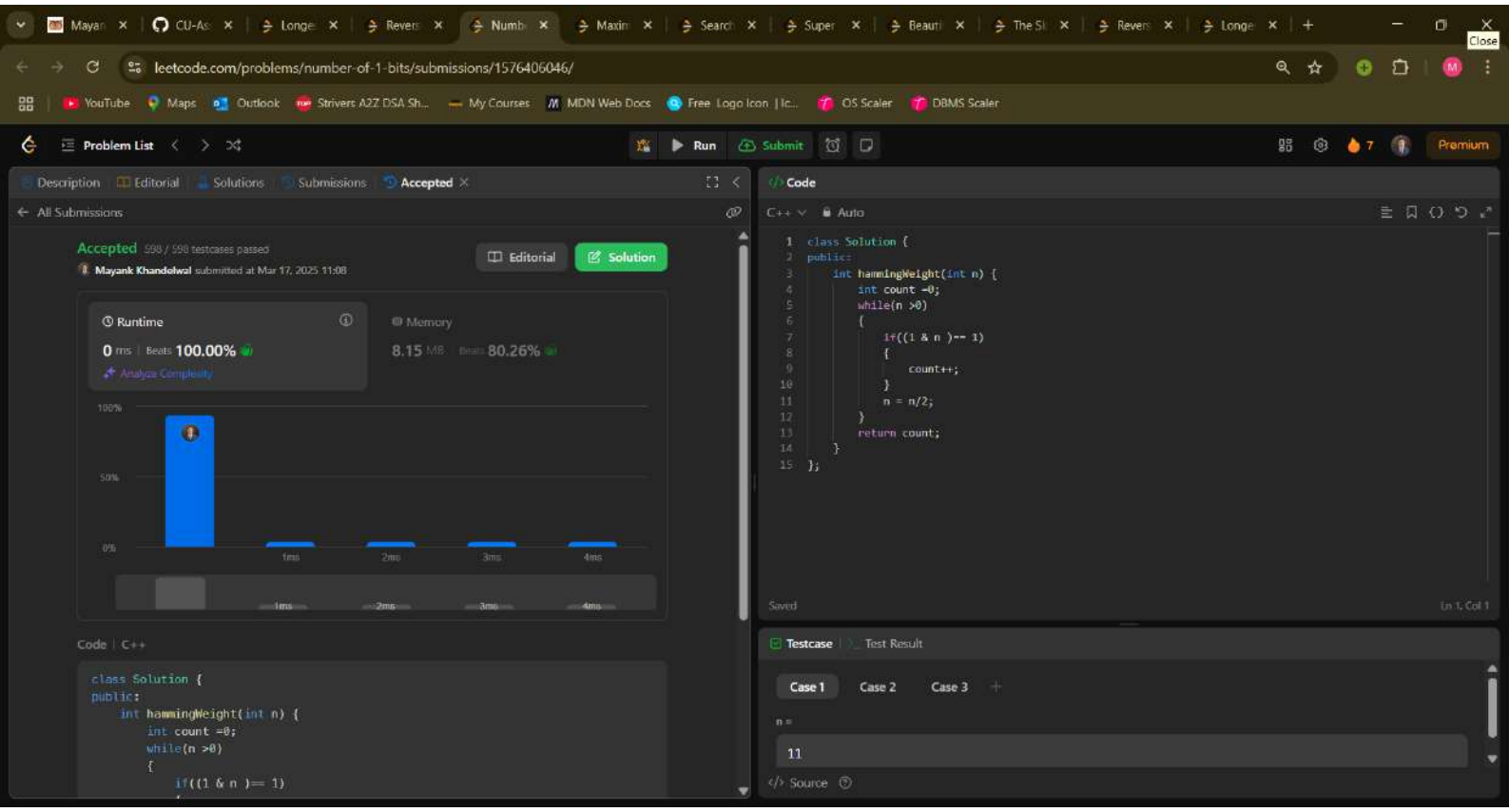
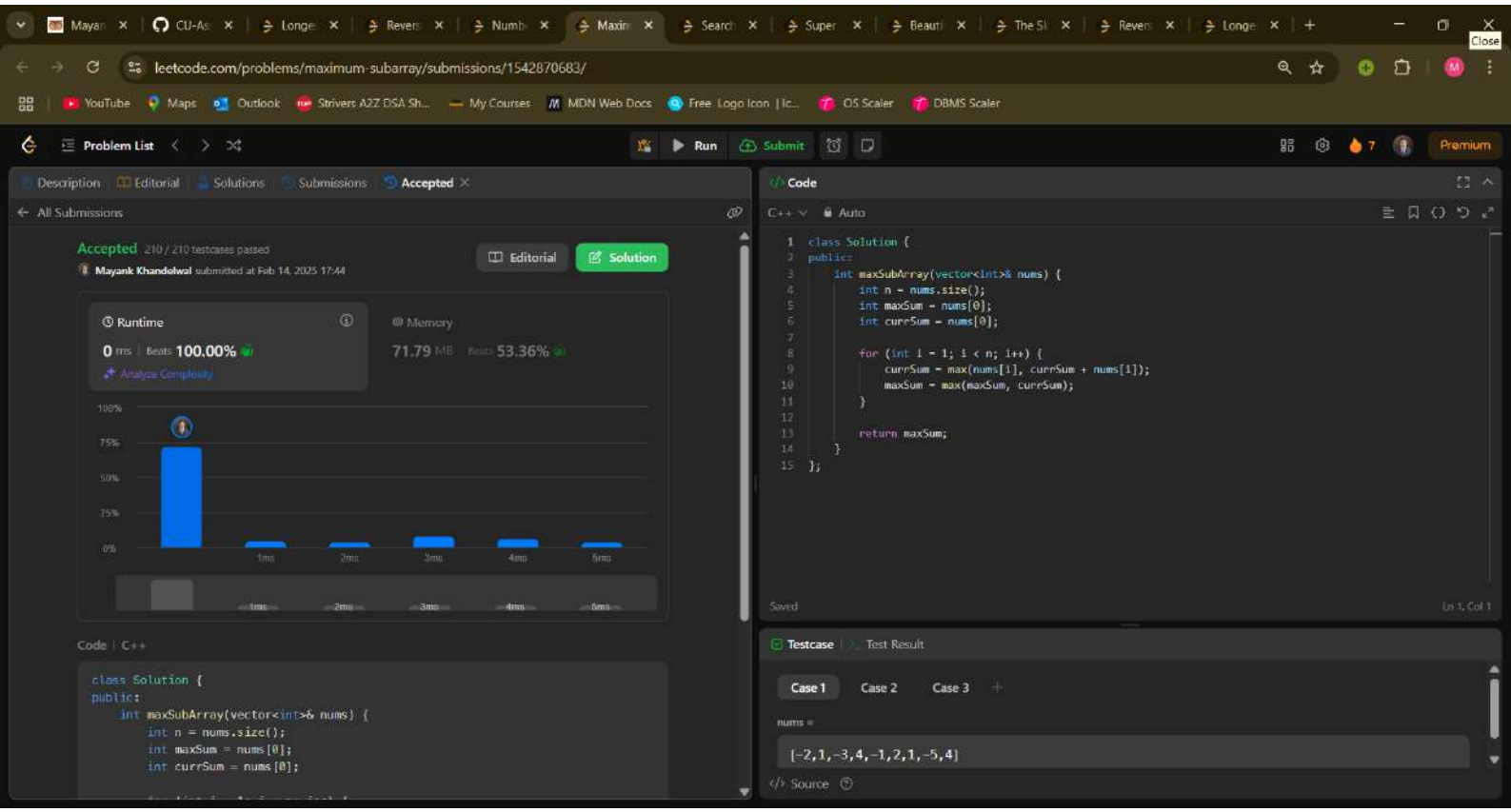
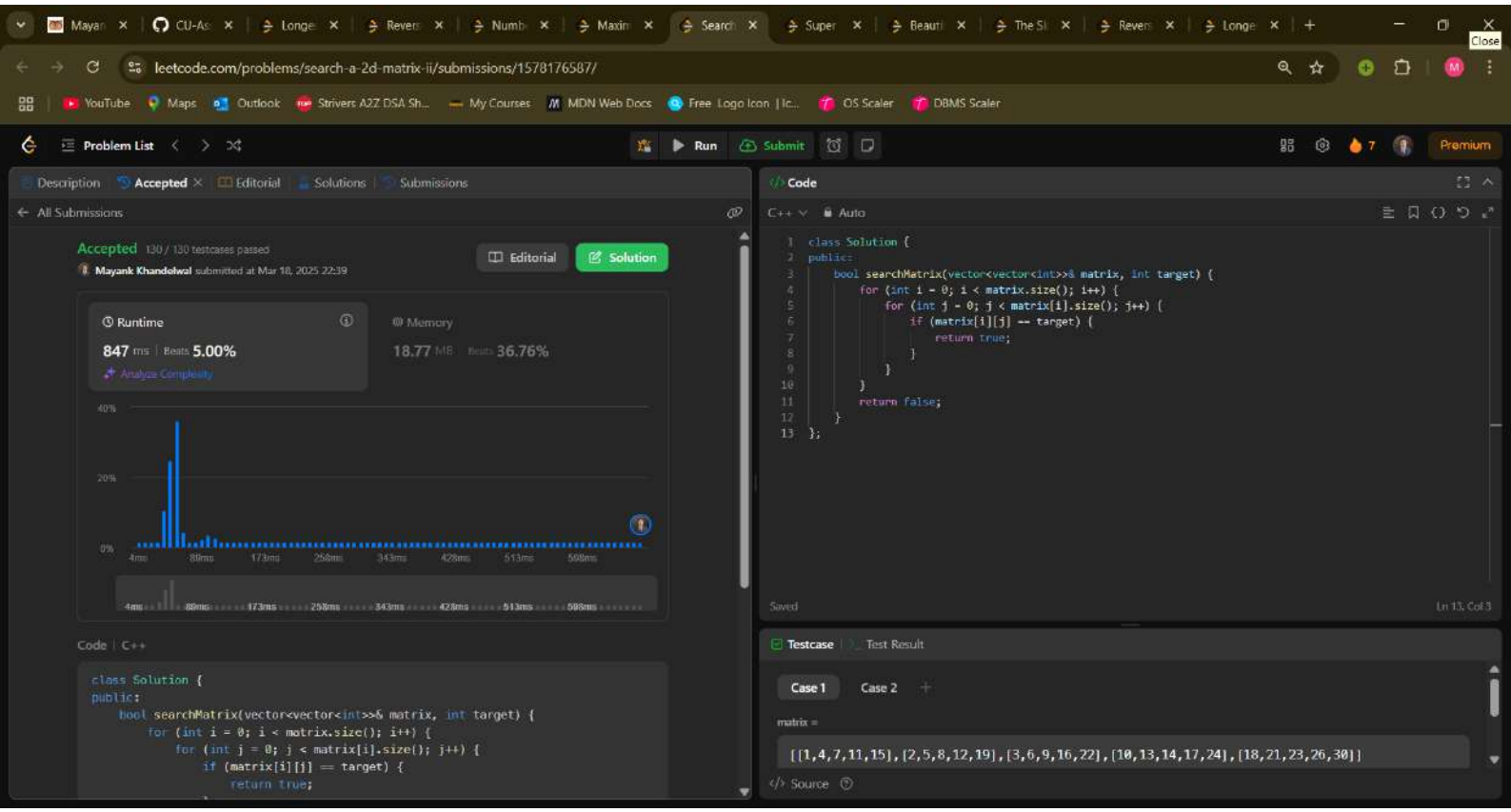
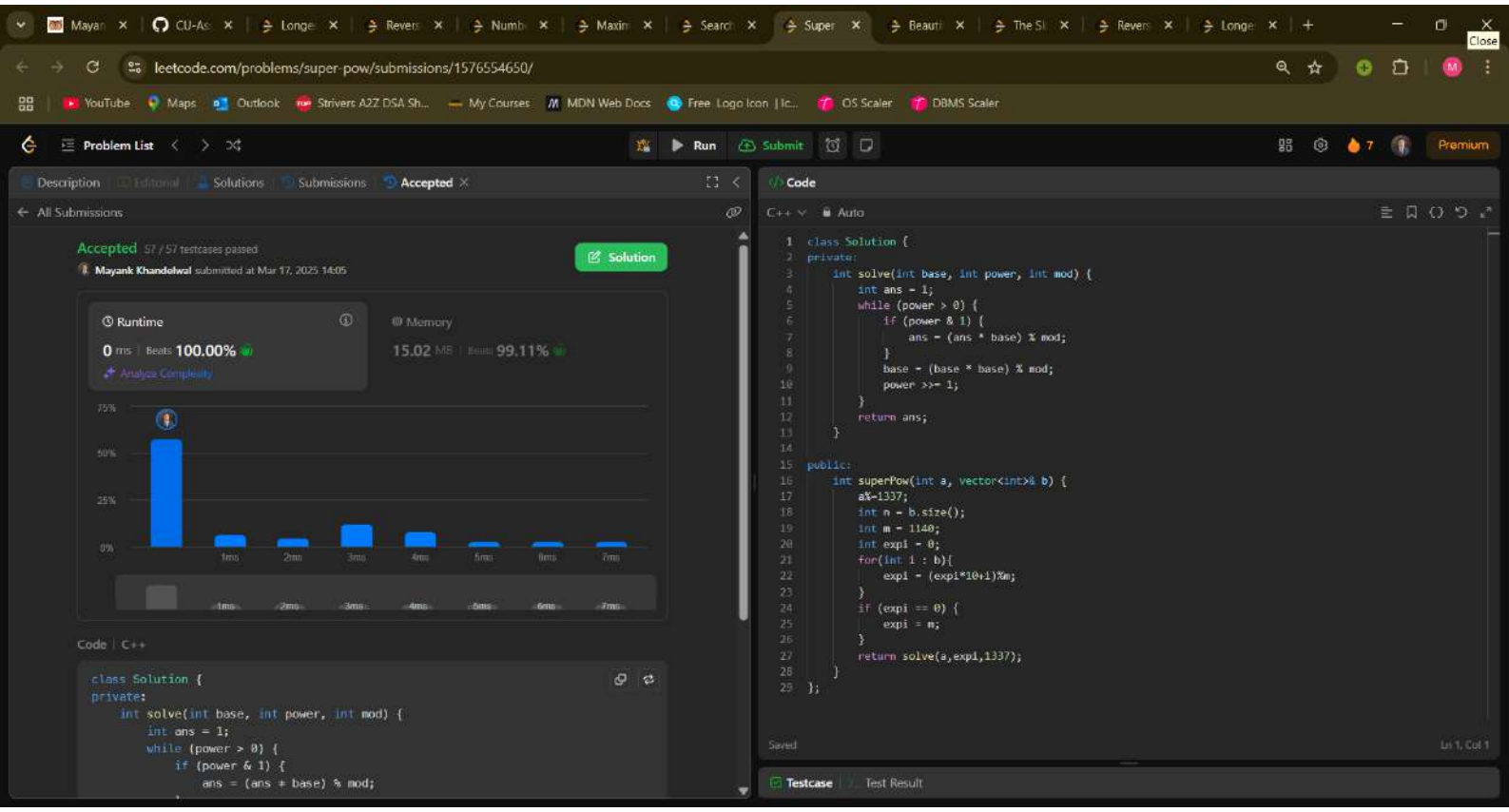

leetcode.com/problems/longest-nice-substring/submissions/1553436284/
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
Accepted
All Submissions
Accepted 73 / 73 testcases passed
Mayank Khandelwal submitted at Feb 24, 2025 10:12
Solution
Runtime
359 ms | Beats 9.46%
Analyze Complexity
Memory
122.60 MB | Beats 9.28%
Analyze Complexity
60%
40%
20%
0%
2ms
48ms
94ms
140ms
186ms
232ms
278ms
325ms
Code | C++
class Solution {
public:
bool isNice(const string &s) {
unordered_set<char> charSet(s.begin(), s.end());
for (char c : s) {
if (charSet.count(tolower(c)) == 0 || charSet.count(toupper(c)) == 0)
return false;
Code
C++
Auto
class Solution {
public:
bool isNice(const string &s) {
unordered_set<char> charSet(s.begin(), s.end());
for (char c : s) {
if (charSet.count(tolower(c)) == 0 || charSet.count(toupper(c)) == 0)
return false;
}
return true;
}
string longestNiceSubstring(string s) {
int n = s.size();
string longest = "";
for (int i = 0; i < n; ++i) {
for (int j = i; j < n; ++j) {
string sub = s.substr(i, j - i + 1);
if (isNice(sub) && sub.length() > longest.length()) {
longest = sub;
}
}
Saved
Ln 2, Col 8
Testcase
Test Result
Case 1
Case 2
Case 3
s =
"YazaAay"
Source

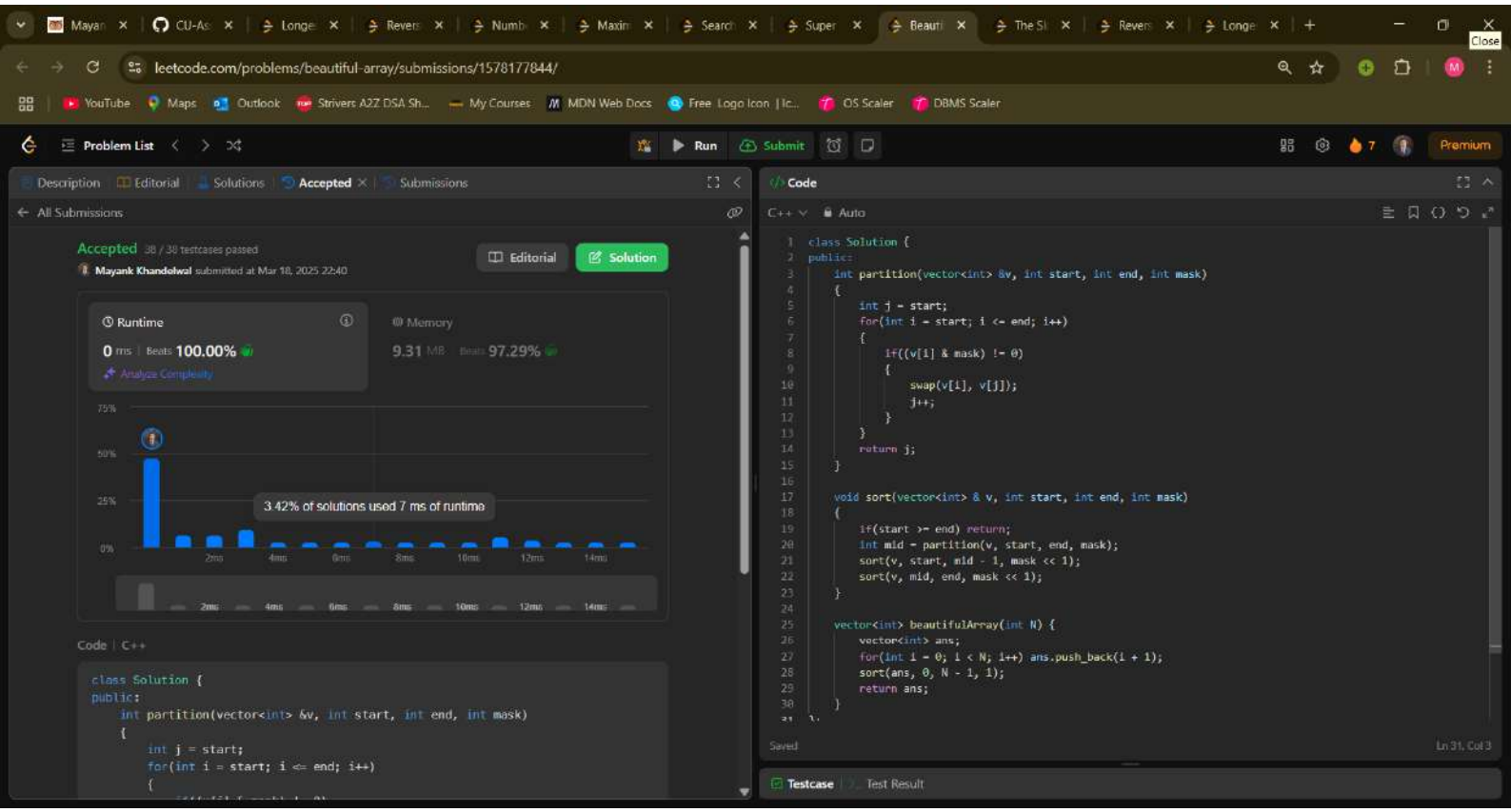

leetcode.com/problems/reverse-bits/submissions/1576446939/
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
Accepted
All Submissions
Accepted 600 / 600 testcases passed
Mayank Khandelwal submitted at Mar 17, 2025 11:51
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
7.62 MB | Beats 87.34%
75%
50%
25%
0%
1ms
2ms
3ms
4ms
Code | C++
class Solution {
public:
uint32_t reverseBits(uint32_t n) {
int ans =0;
for(int i =0;i<32;i++)
{
ans = ans<<1;
Code
C++
Auto
class Solution {
public:
uint32_t reverseBits(uint32_t n) {
int ans =0;
for(int i =0;i<32;i++)
{
ans = ans<<1;
if(n&1)
{
ans = ans | 1 ;
}
n = n>> 1;
}
return ans;
}
};
Saved
Ln 1, Col 1
Testcase
Test Result
Case 1
Case 2
n =
00000010100101000001111010011100
Source


leetcode.com/problems/number-of-1-bits/submissions/1576406046/
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
Accepted
All Submissions
Accepted 598 / 598 testcases passed
Mayank Khandelwal submitted at Mar 17, 2025 11:08
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
8.15 MB | Beats 80.26%
100%
50%
0%
1ms
2ms
3ms
4ms
Code | C++
class Solution {
public:
int hammingWeight(int n) {
int count =0;
while(n >0)
{
if((1 & n)== 1)
Code
C++
Auto
class Solution {
public:
int hammingWeight(int n) {
int count =0;
while(n >0)
{
if((1 & n)== 1)
{
count++;
}
n = n/2;
}
return count;
}
};
Saved
Ln 1, Col 1
Testcase
Test Result
Case 1
Case 2
Case 3
n =
11
Source


leetcode.com/problems/maximum-subarray/submissions/1542870683/
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
Accepted
All Submissions
Accepted 210 / 210 testcases passed
Mayank Khandelwal submitted at Feb 14, 2025 17:44
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
71.79 MB Beats 53.36%
Code | C++
class Solution {
public:
int maxSubArray(vector<int>& nums) {
int n = nums.size();
int maxSum = nums[0];
int currSum = nums[0];
Code
C++
Auto
class Solution {
public:
int maxSubArray(vector<int>& nums) {
int n = nums.size();
int maxSum = nums[0];
int currSum = nums[0];
for (int i = 1; i < n; i++) {
currSum = max(nums[i], currSum + nums[i]);
maxSum = max(maxSum, currSum);
}
return maxSum;
}
};
Saved
Ln 1, Col 1
Testcase
Test Result
Case 1
Case 2
Case 3
nums =
[-2,1,-3,4,-1,2,1,-5,4]
Source


leetcode.com/problems/search-a-2d-matrix-ii/submissions/1578176587/
Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 130 / 130 testcases passed
Mayank Khandelwal submitted at Mar 18, 2025 22:39
Editorial
Solution
Runtime
847 ms | Beats 5.00%
Analyze Complexity
Memory
18.77 MB Beats 36.76%
Code | C++
class Solution {
public:
bool searchMatrix(vector<vector<int>>& matrix, int target) {
for (int i = 0; i < matrix.size(); i++) {
for (int j = 0; j < matrix[i].size(); j++) {
if (matrix[i][j] == target) {
return true;
Code
C++
Auto
class Solution {
public:
bool searchMatrix(vector<vector<int>>& matrix, int target) {
for (int i = 0; i < matrix.size(); i++) {
for (int j = 0; j < matrix[i].size(); j++) {
if (matrix[i][j] == target) {
return true;
}
}
}
return false;
}
};
Saved
Ln 13, Col 3
Testcase
Test Result
Case 1
Case 2
matrix =
[[1,4,7,11,15],[2,5,8,12,19],[3,6,9,16,22],[10,13,14,17,24],[18,21,23,26,30]]
Source

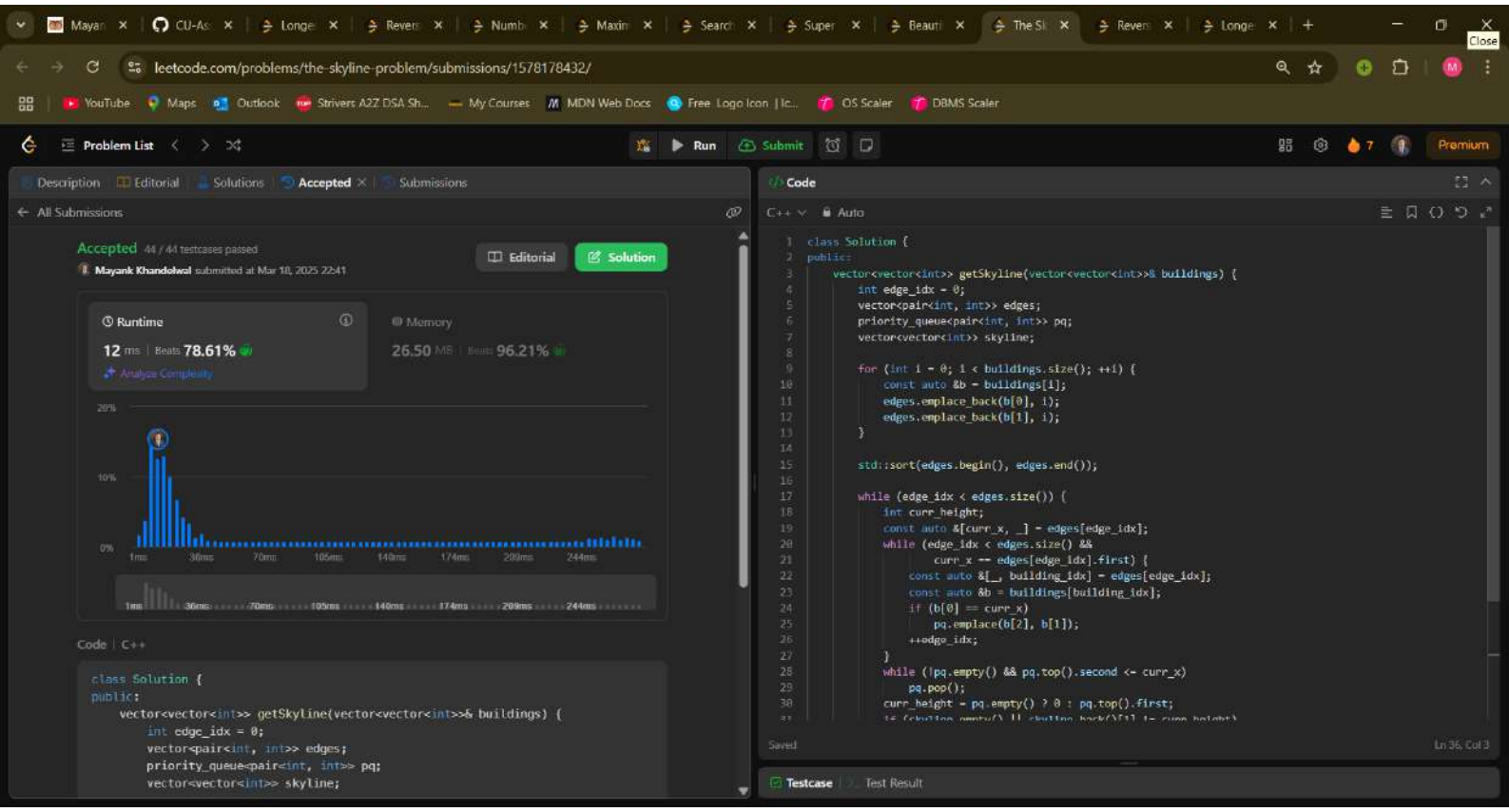

leetcode.com/problems/super-pow/submissions/1576554650/
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
Accepted
All Submissions
Accepted 57 / 57 testcases passed
Mayank Khandelwal submitted at Mar 17, 2025 14:05
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
15.02 MB | Beats 99.11%
Code | C++
class Solution {
private:
int solve(int base, int power, int mod) {
int ans = 1;
while (power > 0) {
if (power & 1) {
ans = (ans * base) % mod;
Code
C++
Auto
class Solution {
private:
int solve(int base, int power, int mod) {
int ans = 1;
while (power > 0) {
if (power & 1) {
ans = (ans * base) % mod;
}
base = (base * base) % mod;
power >>= 1;
}
return ans;
}
public:
int superPow(int a, vector<int>& b) {
a%=1337;
int n = b.size();
int m = 1140;
int expi = 0;
for(int i : b){
expi = (expi*10+i)%m;
}
if (expi == 0) {
expi = m;
}
return solve(a,expi,1337);
}
};
Saved
Ln 1, Col 1
Testcase
Test Result




```cpp
class Solution {
public:
    int partition(vector<int> &v, int start, int end, int mask)
    {
        int j = start;
        for(int i = start; i <= end; i++)
        {
            if((v[i] & mask) != 0)
            {
                swap(v[i], v[j]);
                j++;
            }
        }
        return j;
    }

    void sort(vector<int> & v, int start, int end, int mask)
    {
        if(start >= end) return;
        int mid = partition(v, start, end, mask);
        sort(v, start, mid - 1, mask << 1);
        sort(v, mid, end, mask << 1);
    }

    vector<int> beautifulArray(int N) {
        vector<int> ans;
        for(int i = 0; i < N; i++) ans.push_back(i + 1);
        sort(ans, 0, N - 1, 1);
        return ans;
    }
};
```

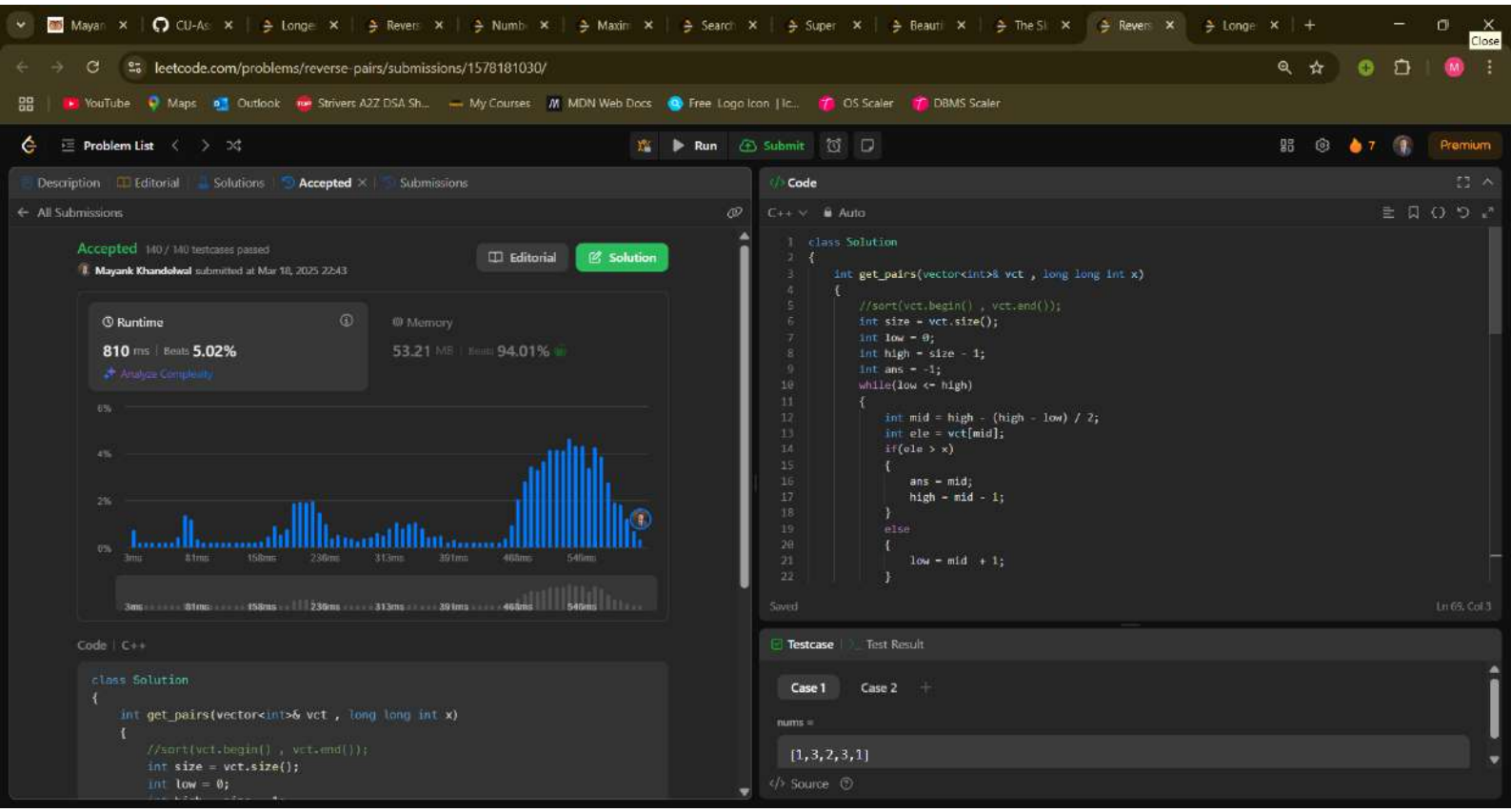


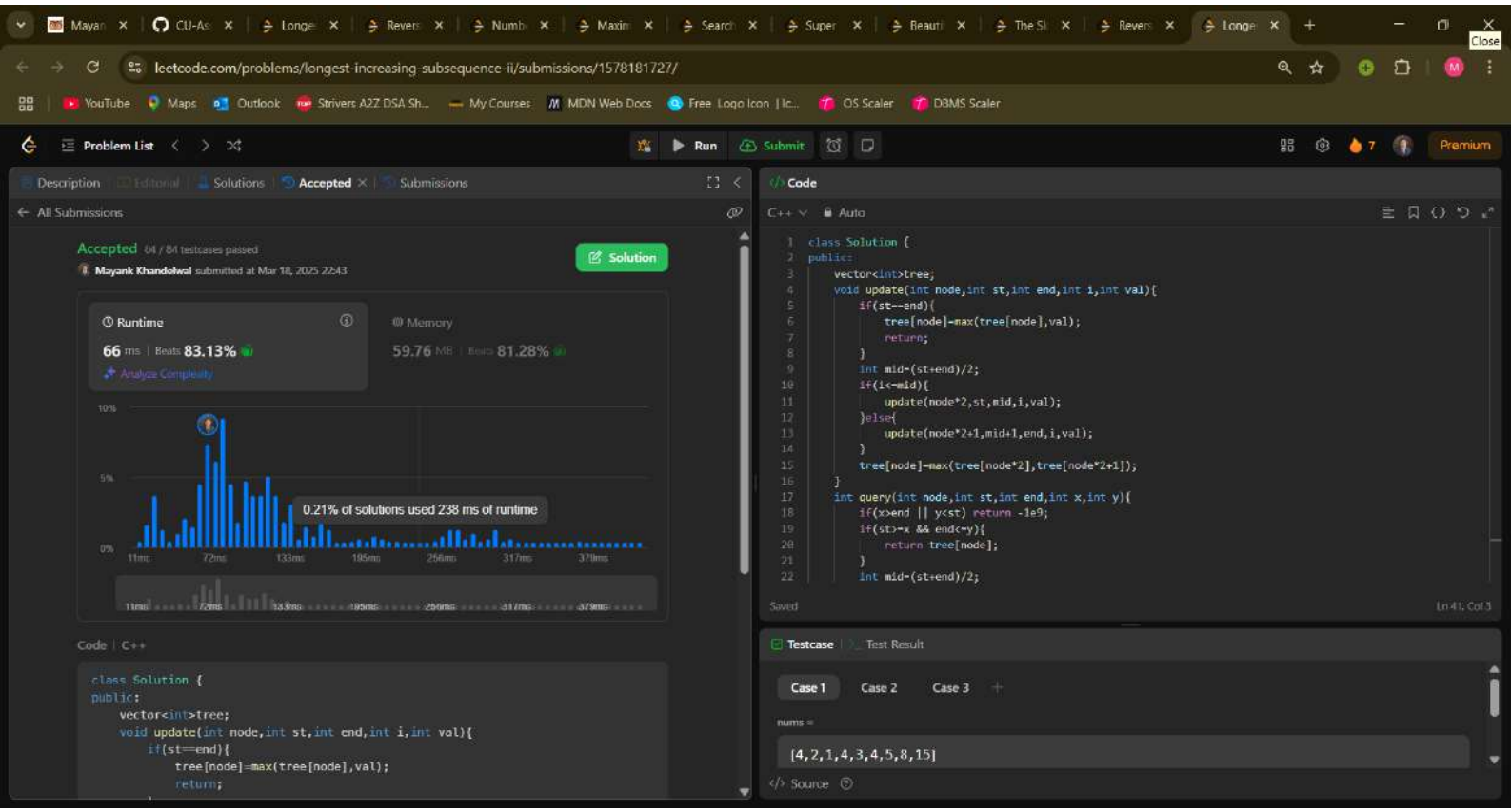

```cpp
class Solution {
public:
    vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
        int edge_idx = 0;
        vector<pair<int, int>> edges;
        priority_queue<pair<int, int>> pq;
        vector<vector<int>> skyline;

        for (int i = 0; i < buildings.size(); ++i) {
            const auto &b = buildings[i];
            edges.emplace_back(b[0], i);
            edges.emplace_back(b[1], i);
        }

        std::sort(edges.begin(), edges.end());

        while (edge_idx < edges.size()) {
            int curr_height;
            const auto &[curr_x, _] = edges[edge_idx];
            while (edge_idx < edges.size() &&
                   curr_x == edges[edge_idx].first) {
                const auto &[_, building_idx] = edges[edge_idx];
                const auto &b = buildings[building_idx];
                if (b[0] == curr_x)
                    pq.emplace(b[2], b[1]);
                ++edge_idx;
            }
            while (!pq.empty() && pq.top().second <= curr_x)
                pq.pop();
            curr_height = pq.empty() ? 0 : pq.top().first;
            if (skyline.empty() || skyline.back()[1] != curr_height)
```


leetcode.com/problems/reverse-pairs/submissions/1578181030/
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Accepted
Submissions
All Submissions
Accepted 140 / 140 testcases passed
Mayank Khandelwal submitted at Mar 18, 2025 22:43
Editorial
Solution
Runtime
810 ms | Beats 5.02%
Analyze Complexity
Memory
53.21 MB | Beats 94.01%
Code | C++
class Solution
{
int get_pairs(vector<int>& vct , long long int x)
{
//sort(vct.begin() , vct.end());
int size = vct.size();
int low = 0;
Code
C++
Auto
class Solution
{
int get_pairs(vector<int>& vct , long long int x)
{
//sort(vct.begin() , vct.end());
int size = vct.size();
int low = 0;
int high = size - 1;
int ans = -1;
while(low <= high)
{
int mid = high - (high - low) / 2;
int ele = vct[mid];
if(ele > x)
{
ans = mid;
high = mid - 1;
}
else
{
low = mid + 1;
}
Saved
Ln 69, Col 3
Testcase
Test Result
Case 1
Case 2
nums =
[1,3,2,3,1]
Source




```cpp
class Solution {
public:
    vector<int>tree;
    void update(int node,int st,int end,int i,int val){
        if(st==end){
            tree[node]=max(tree[node],val);
            return;
        }
        int mid=(st+end)/2;
        if(i<=mid){
            update(node*2,st,mid,i,val);
        }else{
            update(node*2+1,mid+1,end,i,val);
        }
        tree[node]=max(tree[node*2],tree[node*2+1]);
    }
    int query(int node,int st,int end,int x,int y){
        if(x>end || y<st) return -1e9;
        if(st>=x && end<=y){
            return tree[node];
        }
        int mid=(st+end)/2;
```