Advanced Programming

Assignment 4

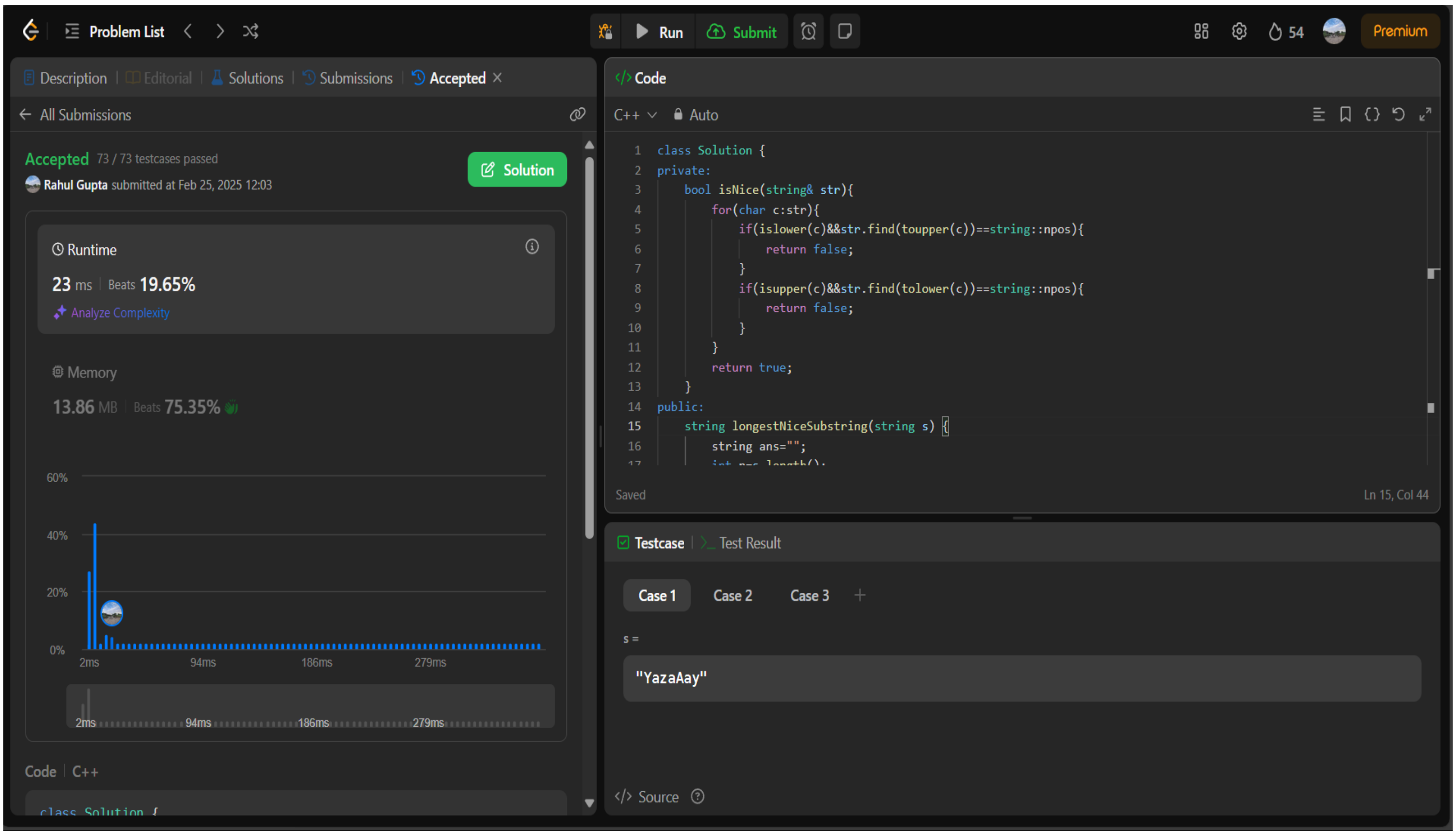
Name-Rahul Gupta

UID-22BCS10469

Q1.) Longest Nice Substring

```
class Solution {
private:
    bool isNice(string& str){
        for(char c:str){
            if(islower(c)&&str.find(toupper(c))==string::npos){
                return false;
            }
            if(isupper(c)&&str.find(tolower(c))==string::npos){
                return false;
            }
        }
        return true;
    }
public:
    string longestNiceSubstring(string s) {
        string ans="";
        int n=s.length();
        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)){
                    if(sub.length()>ans.length()){
```

```
                    ans=sub;
                }
            }
        }
    }
    return ans;
  }
};
```



Accepted 73 / 73 testcases passed

Rahul Gupta submitted at Feb 25, 2025 12:03

Runtime

23 ms | Beats 19.65%

Memory

13.86 MB | Beats 75.35%

Code | C++

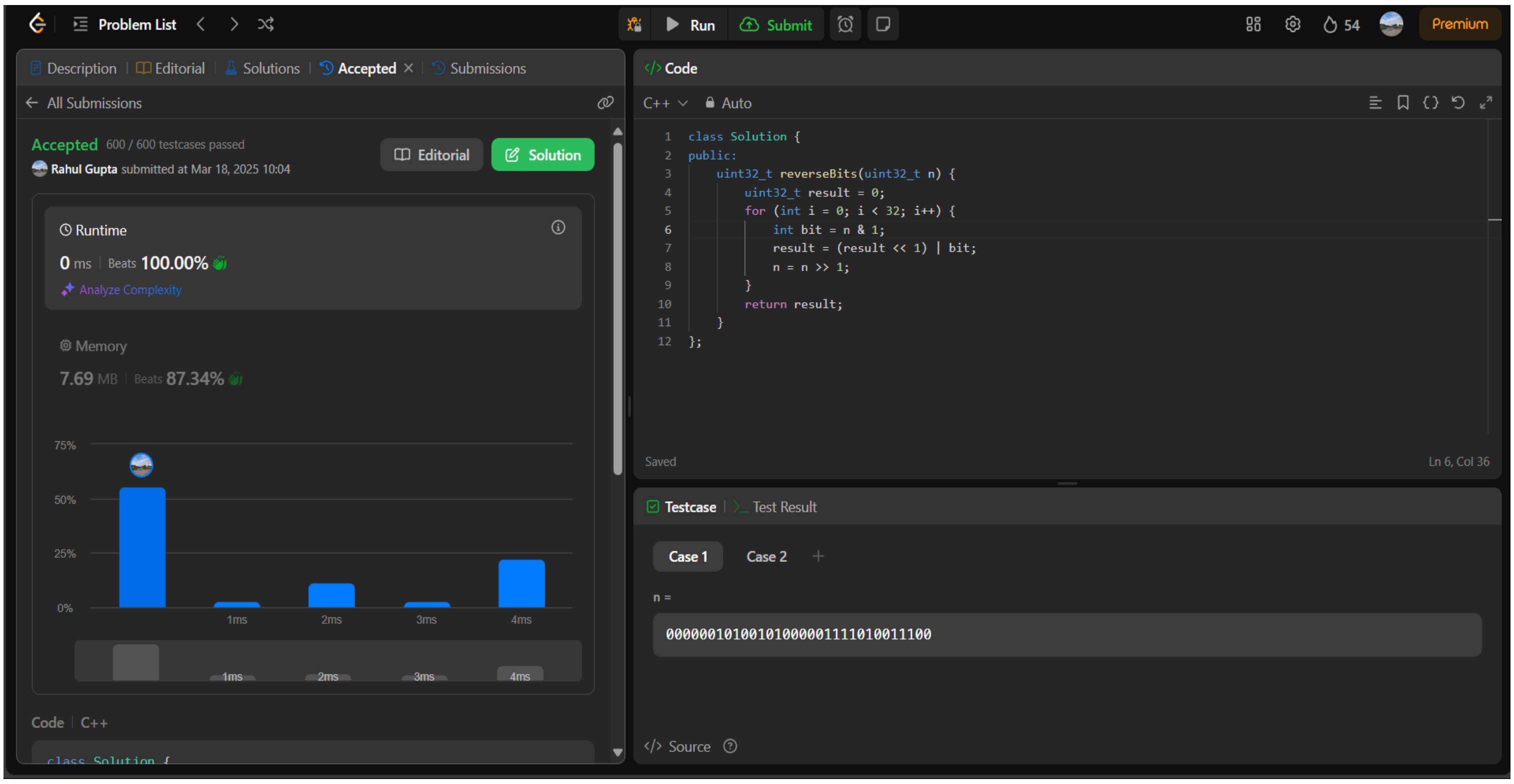
```
class Solution {
private:
    bool isNice(string& str){
        for(char c:str){
            if(islower(c)&&str.find(toupper(c))==string::npos){
                return false;
            }
            if(isupper(c)&&str.find(tolower(c))==string::npos){
                return false;
            }
        }
        return true;
    }
public:
    string longestNiceSubstring(string s) {
        string ans="";
```

s =

"YazaAay"

Q2.) Reverse Bits

```
class Solution {
public:
    uint32_t reverseBits(uint32_t n) {
        uint32_t result = 0;
        for (int i = 0; i < 32; i++) {
            int bit = n & 1;
            result = (result << 1) | bit;
            n = n >> 1;
        }
        return result;
    }
};
```



Accepted 600 / 600 testcases passed
Rahul Gupta submitted at Mar 18, 2025 10:04

Runtime
0 ms | Beats 100.00%

Memory
7.69 MB | Beats 87.34%

```
class Solution {
public:
    uint32_t reverseBits(uint32_t n) {
        uint32_t result = 0;
        for (int i = 0; i < 32; i++) {
            int bit = n & 1;
            result = (result << 1) | bit;
            n = n >> 1;
        }
        return result;
    }
};
```

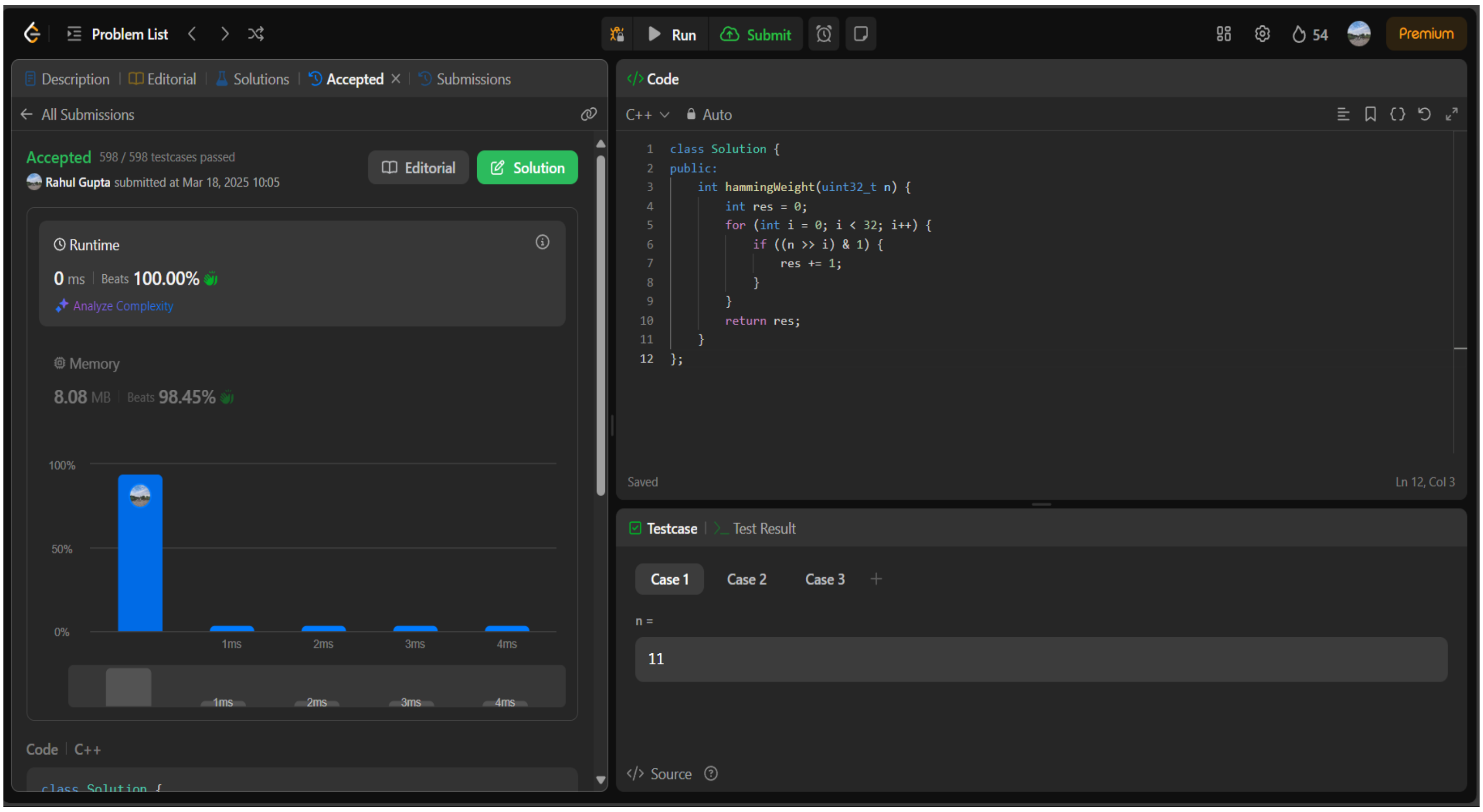
n =

```
00000010100101000001111010011100
```

Code | C++

Q3.) Number of 1 Bits

```
class Solution {
public:
    int hammingWeight(uint32_t n) {
        int res = 0;
        for (int i = 0; i < 32; i++) {
            if ((n >> i) & 1) {
                res += 1;
            }
        }
        return res;
    }
};
```



Accepted 598 / 598 testcases passed

Rahul Gupta submitted at Mar 18, 2025 10:05

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

8.08 MB | Beats 98.45%

Code | C++

```
class Solution {
public:
    int hammingWeight(uint32_t n) {
        int res = 0;
        for (int i = 0; i < 32; i++) {
            if ((n >> i) & 1) {
                res += 1;
            }
        }
        return res;
    }
};
```

Saved

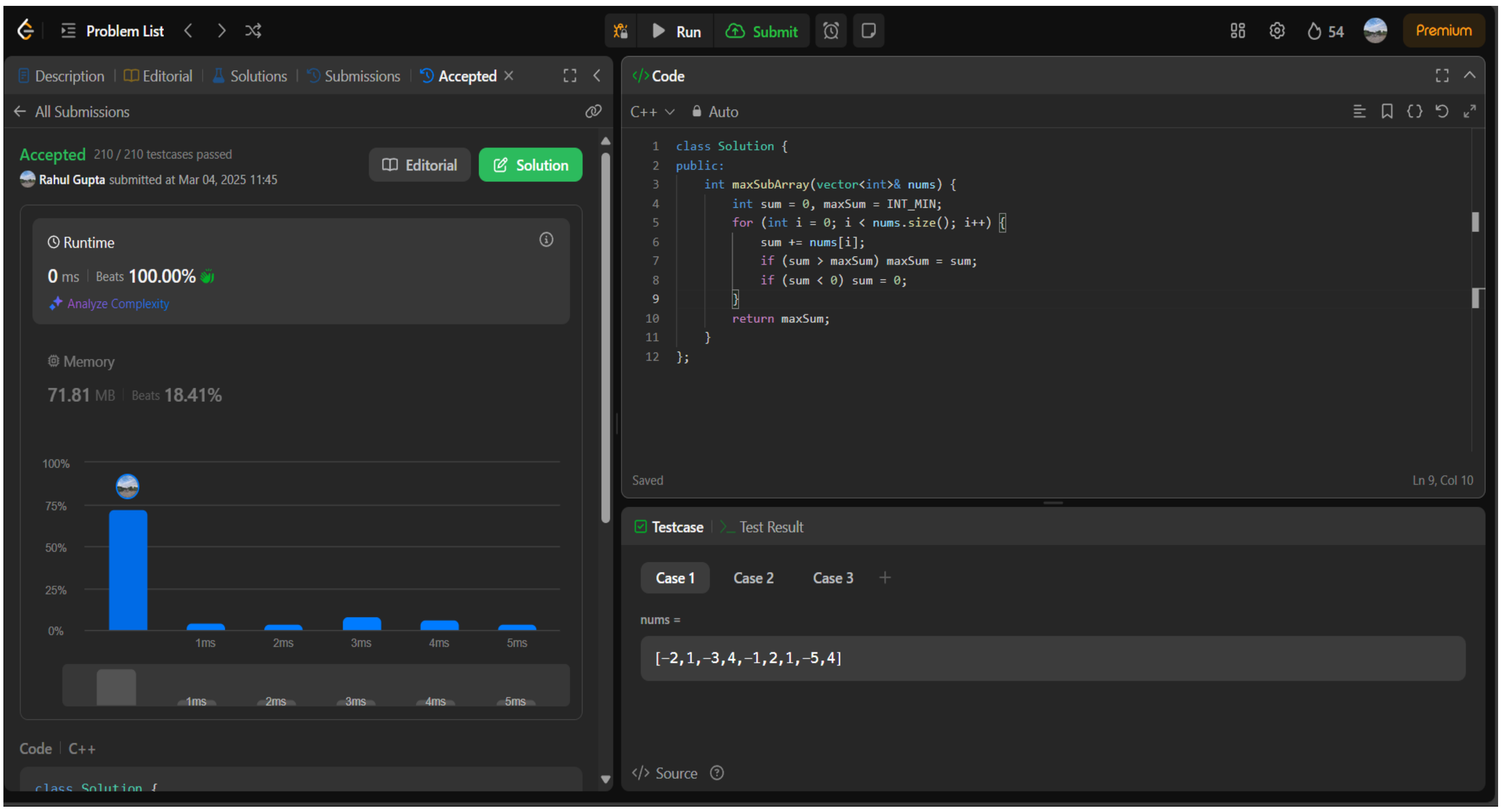
Ln 12, Col 3

n =

11

Q4.) Maximum Subarray

```
class Solution {

public:

   int maxSubArray(vector<int>& nums) {

      int sum = 0, maxSum = INT_MIN;

      for (int i = 0; i < nums.size(); i++) {

         sum += nums[i];

         if (sum > maxSum) maxSum = sum;

         if (sum < 0) sum = 0;

      }

      return maxSum;

   }

};
```



Accepted 210 / 210 testcases passed

Rahul Gupta submitted at Mar 04, 2025 11:45

Runtime

0 ms | Beats 100.00%

Memory

71.81 MB | Beats 18.41%

Code | C++

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int sum = 0, maxSum = INT_MIN;
        for (int i = 0; i < nums.size(); i++) {
            sum += nums[i];
            if (sum > maxSum) maxSum = sum;
            if (sum < 0) sum = 0;
        }
        return maxSum;
    }
};
```

Saved

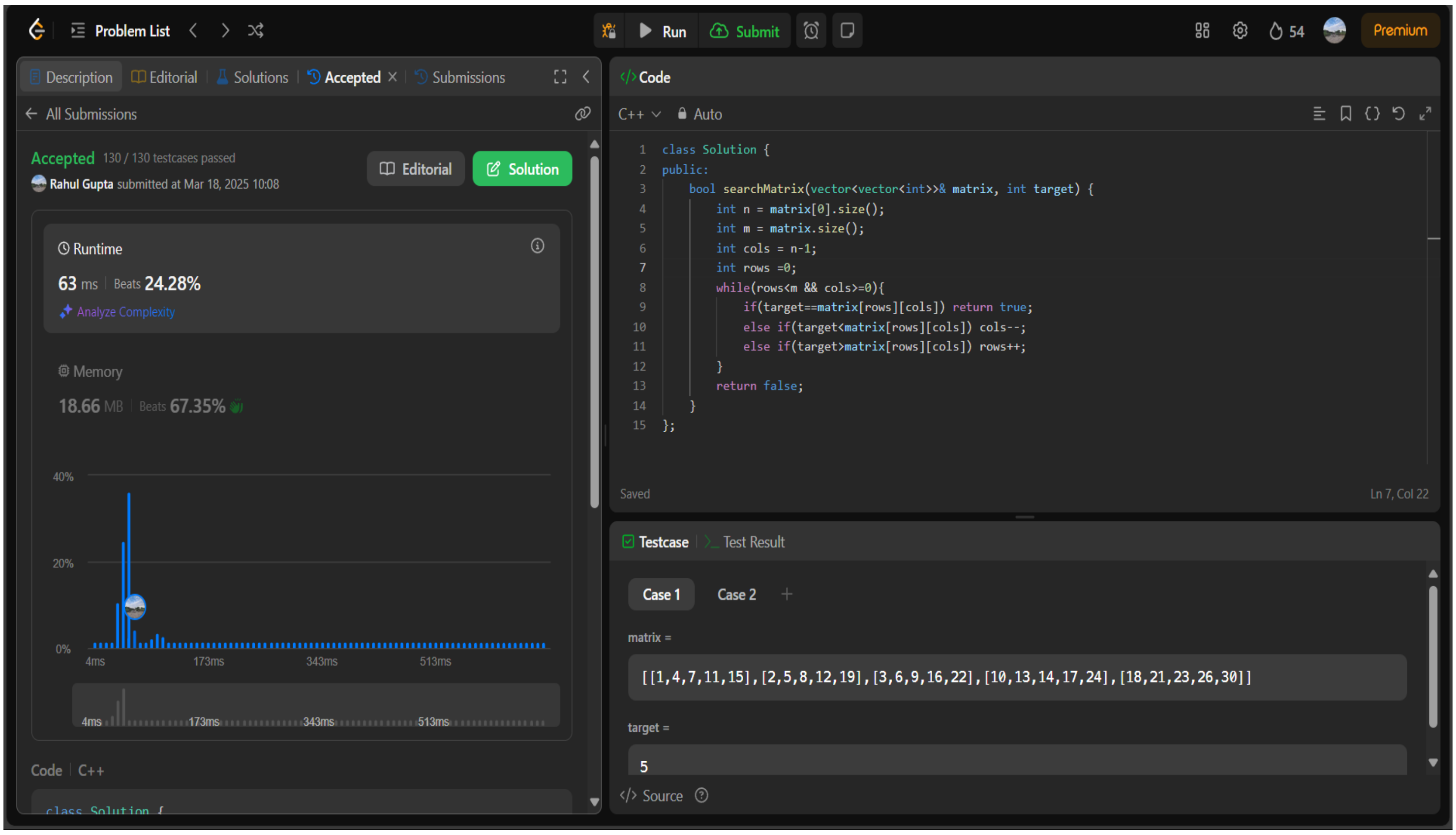
Ln 9, Col 10

nums =

```
[-2,1,-3,4,-1,2,1,-5,4]
```

Q5.) Search a 2D Matrix II

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int n = matrix[0].size();
        int m = matrix.size();
        int cols = n-1;
        int rows =0;
        while(rows<m && cols>=0){
            if(target==matrix[rows][cols]) return true;
            else if(target<matrix[rows][cols]) cols--;
            else if(target>matrix[rows][cols]) rows++;
        }
        return false;
    }
};
```



Accepted 130 / 130 testcases passed
Rahul Gupta submitted at Mar 18, 2025 10:08

Runtime
63 ms | Beats 24.28%

Memory
18.66 MB | Beats 67.35%

Code | C++

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int n = matrix[0].size();
        int m = matrix.size();
        int cols = n-1;
        int rows =0;
        while(rows<m && cols>=0){
            if(target==matrix[rows][cols]) return true;
            else if(target<matrix[rows][cols]) cols--;
            else if(target>matrix[rows][cols]) rows++;
        }
        return false;
    }
};
```

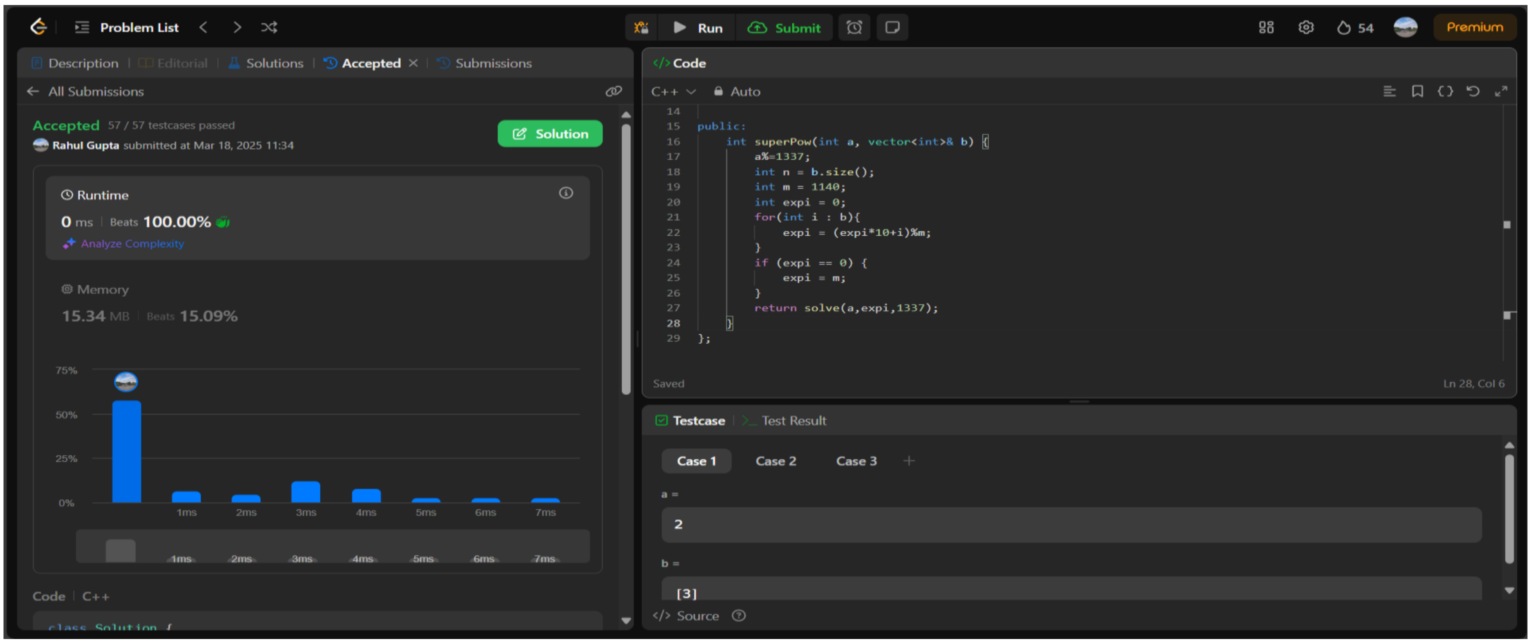
Saved
Ln 7, Col 22

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]]

target =
5

Q6.) Super Pow

```
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); }};
```



Accepted 57 / 57 testcases passed
Rahul Gupta submitted at Mar 18, 2025 11:34

Runtime
0 ms | Beats 100.00%

Memory
15.34 MB | Beats 15.09%

Code | C++

```
class Solution {
```

```
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);
    }
};
```

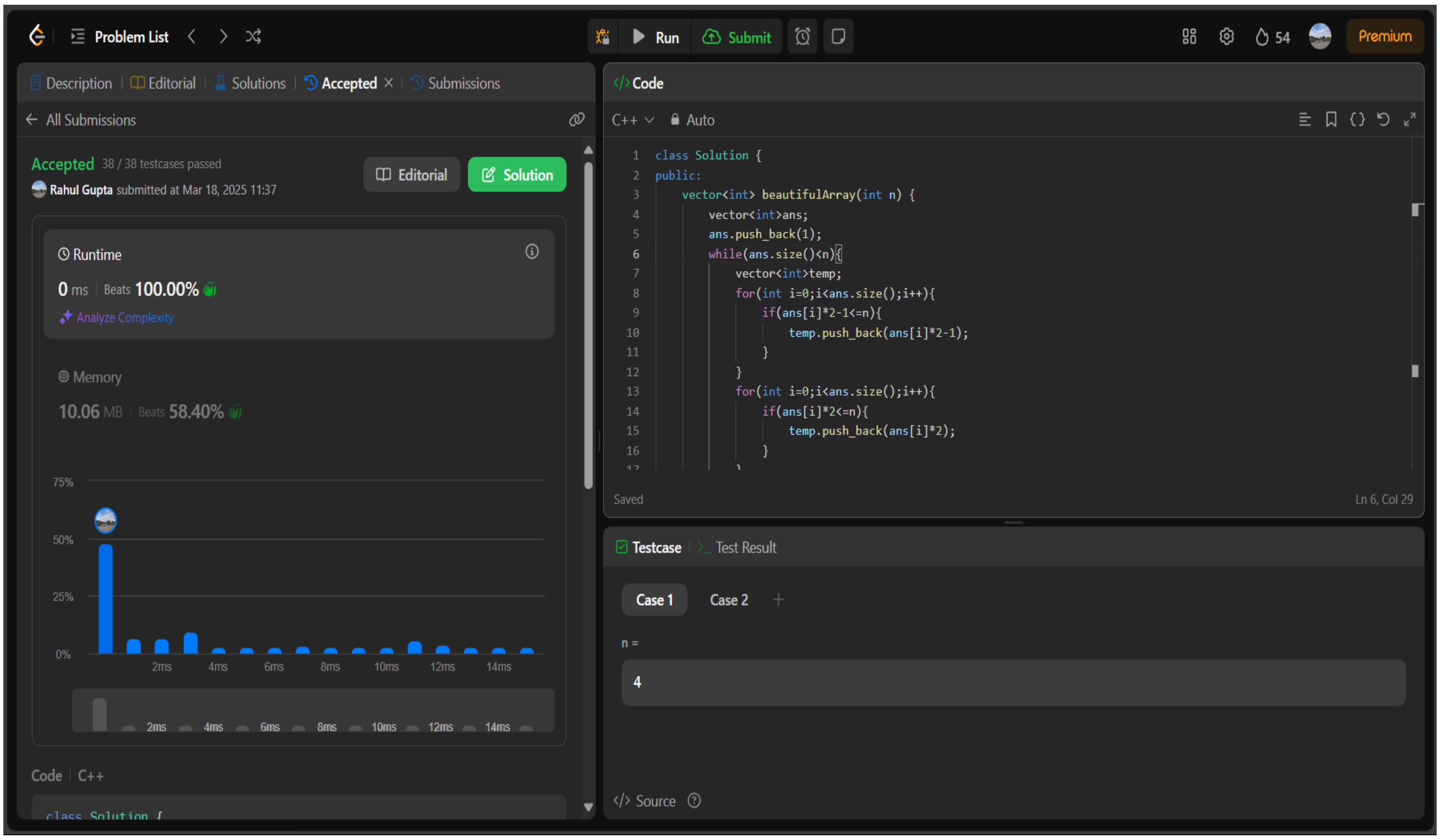
Ln 28, Col 6

a =
2

b =
[3]

Q7.) Beautiful Array

```
class Solution {
public:
    vector<int> beautifulArray(int n) {
        vector<int>ans;
        ans.push_back(1);
        while(ans.size()<n){
            vector<int>temp;
            for(int i=0;i<ans.size();i++){
                if(ans[i]*2-1<=n){
                    temp.push_back(ans[i]*2-1); } }
            for(int i=0;i<ans.size();i++){
                if(ans[i]*2<=n){
                    temp.push_back(ans[i]*2); }}
            ans=temp; }
        return ans; } };
```



Accepted 38 / 38 testcases passed

Rahul Gupta submitted at Mar 18, 2025 11:37

Runtime

0 ms | Beats 100.00%

Memory

10.06 MB | Beats 58.40%

Code | C++

```
class Solution {
public:
    vector<int> beautifulArray(int n) {
        vector<int>ans;
        ans.push_back(1);
        while(ans.size()<n){
            vector<int>temp;
            for(int i=0;i<ans.size();i++){
                if(ans[i]*2-1<=n){
                    temp.push_back(ans[i]*2-1);
                }
            }
            for(int i=0;i<ans.size();i++){
                if(ans[i]*2<=n){
                    temp.push_back(ans[i]*2);
                }
```

Saved

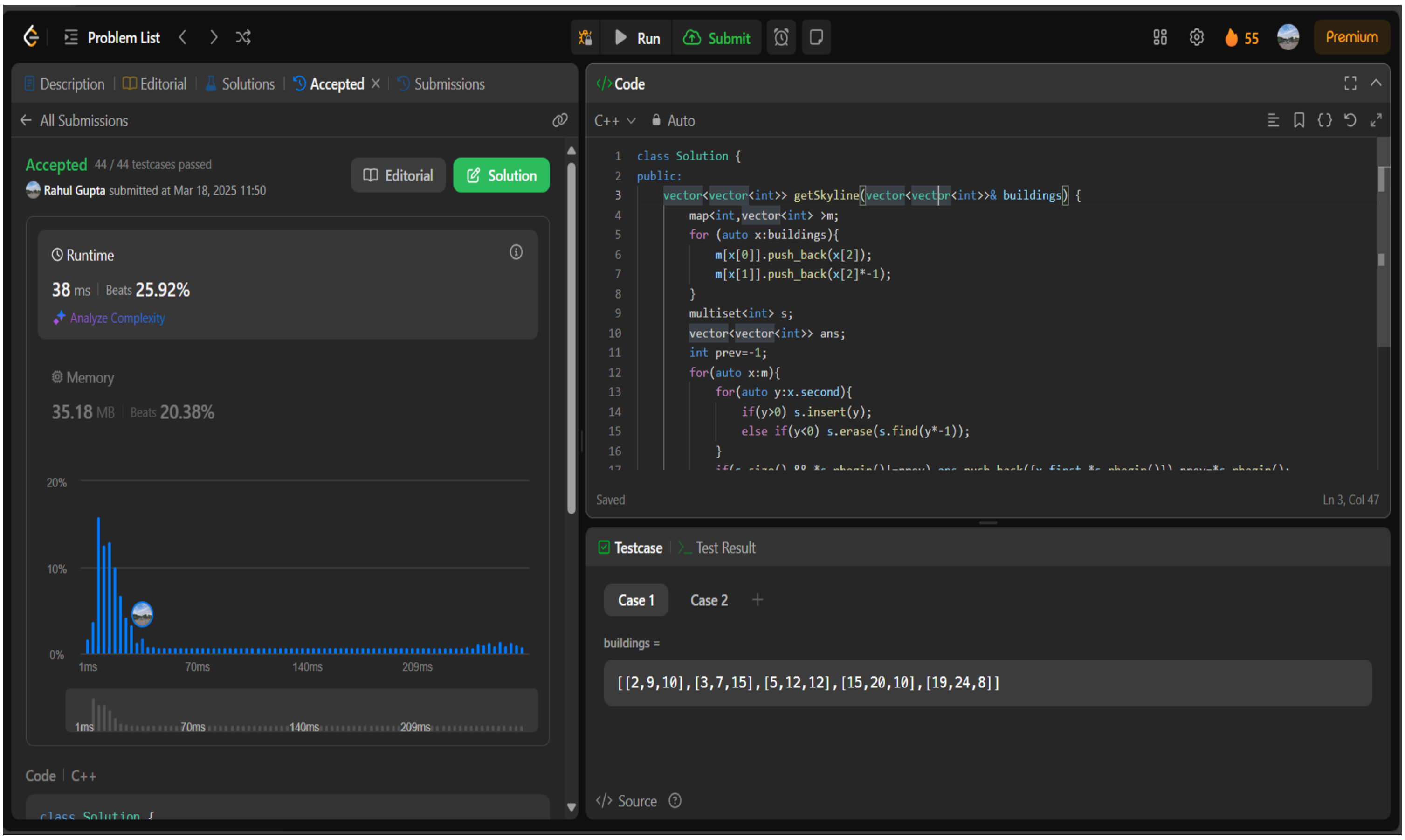
Ln 6, Col 29

n =

4

Q8.) The Skyline Problem

```cpp
class Solution {

public:

    vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {

        map<int,vector<int> >m;

        for (auto x:buildings){

            m[x[0]].push_back(x[2]);

            m[x[1]].push_back(x[2]*-1); }

        multiset<int> s;

        vector<vector<int>> ans;

        int prev=-1;

        for(auto x:m){

            for(auto y:x.second){

                if(y>0) s.insert(y);

                else if(y<0) s.erase(s.find(y*-1)); }

            if(s.size() && *s.rbegin()!=prev) ans.push_back({x.first,*s.rbegin()}),prev=*s.rbegin();

            if(!s.size()) ans.push_back({x.first,0}),prev=0;   }

        return ans; } };
```



Accepted 44 / 44 testcases passed
Rahul Gupta submitted at Mar 18, 2025 11:50

Runtime
38 ms | Beats 25.92%

Memory
35.18 MB | Beats 20.38%

Code | C++

```cpp
class Solution {
public:
    vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
        map<int,vector<int> >m;
        for (auto x:buildings){
            m[x[0]].push_back(x[2]);
            m[x[1]].push_back(x[2]*-1);
        }
        multiset<int> s;
        vector<vector<int>> ans;
        int prev=-1;
        for(auto x:m){
            for(auto y:x.second){
                if(y>0) s.insert(y);
                else if(y<0) s.erase(s.find(y*-1));
            }
```

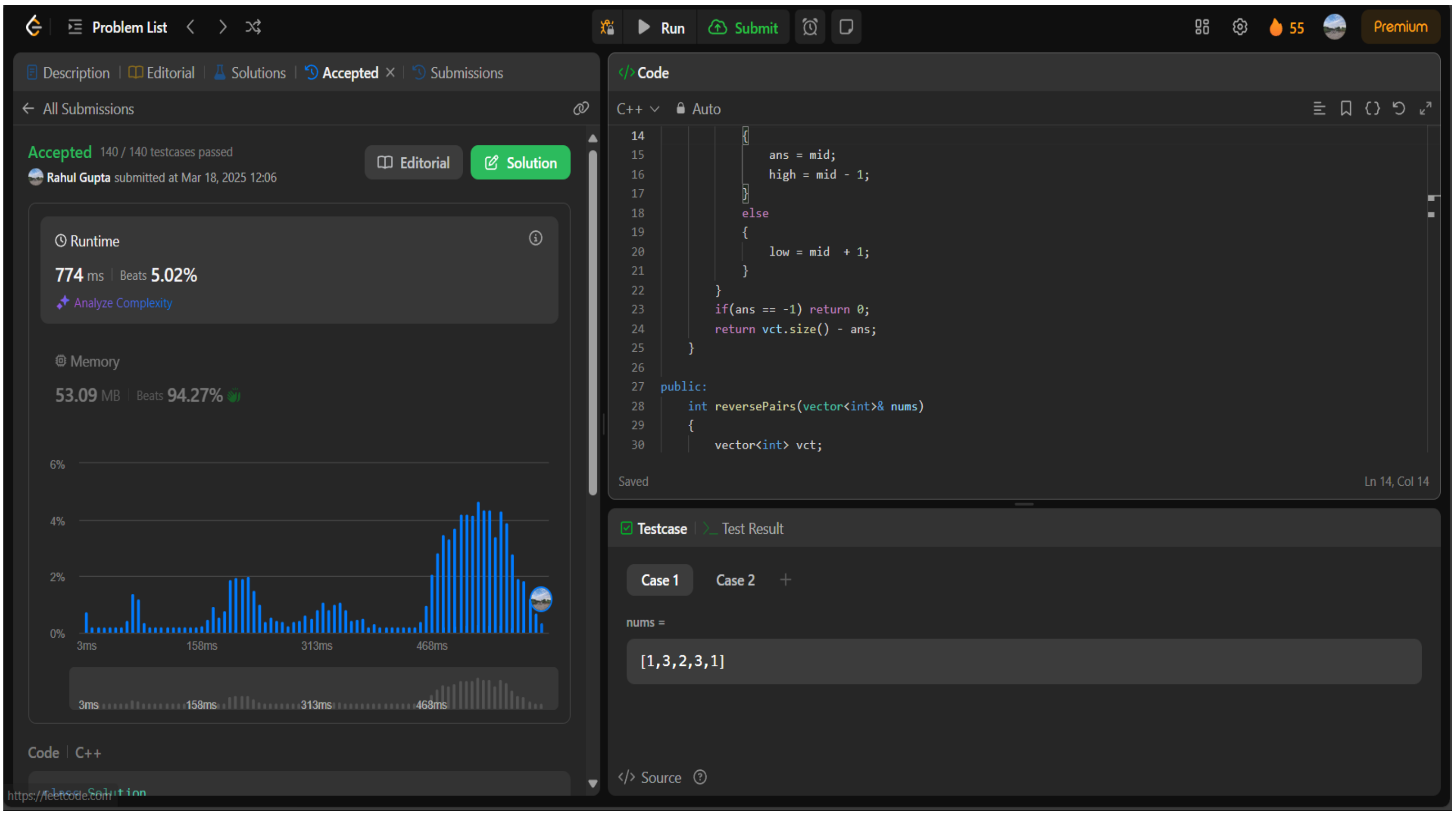
Saved   Ln 3, Col 47

buildings =

```
[[2,9,10],[3,7,15],[5,12,12],[15,20,10],[19,24,8]]
```

Q9.) Reverse Pairs

```
class Solution  {
    int get_pairs(vector<int>& vct , long long int x) {
        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; }}
        if(ans == -1) return 0;
        return vct.size() - ans; }
public:
    int reversePairs(vector<int>& nums) {
        vector<int> vct;
        int counter = 0;
        for(auto it : nums) {
            long long int x = 1LL * 2 * it;
            counter += get_pairs(vct , x);
            int low = 0;
            int high = vct.size();
            int ans = vct.size();
            while(low < high) {
                int mid = low + (high - low) / 2;
                if(vct[mid] >= it) {
                    ans = mid;
                    high = mid; }
                else {
                    low = mid + 1; }}
            vct.insert(vct.begin() + ans , it); }
        return counter; }
};
```



Accepted 140 / 140 testcases passed

Rahul Gupta submitted at Mar 18, 2025 12:06

Runtime

774 ms | Beats 5.02%

Memory

53.09 MB | Beats 94.27%

Code | C++

```
            {
                ans = mid;
                high = mid - 1;
            }
            else
            {
                low = mid  + 1;
            }
        }
        if(ans == -1) return 0;
        return vct.size() - ans;
    }

public:
    int reversePairs(vector<int>& nums)
    {
        vector<int> vct;
```

Saved

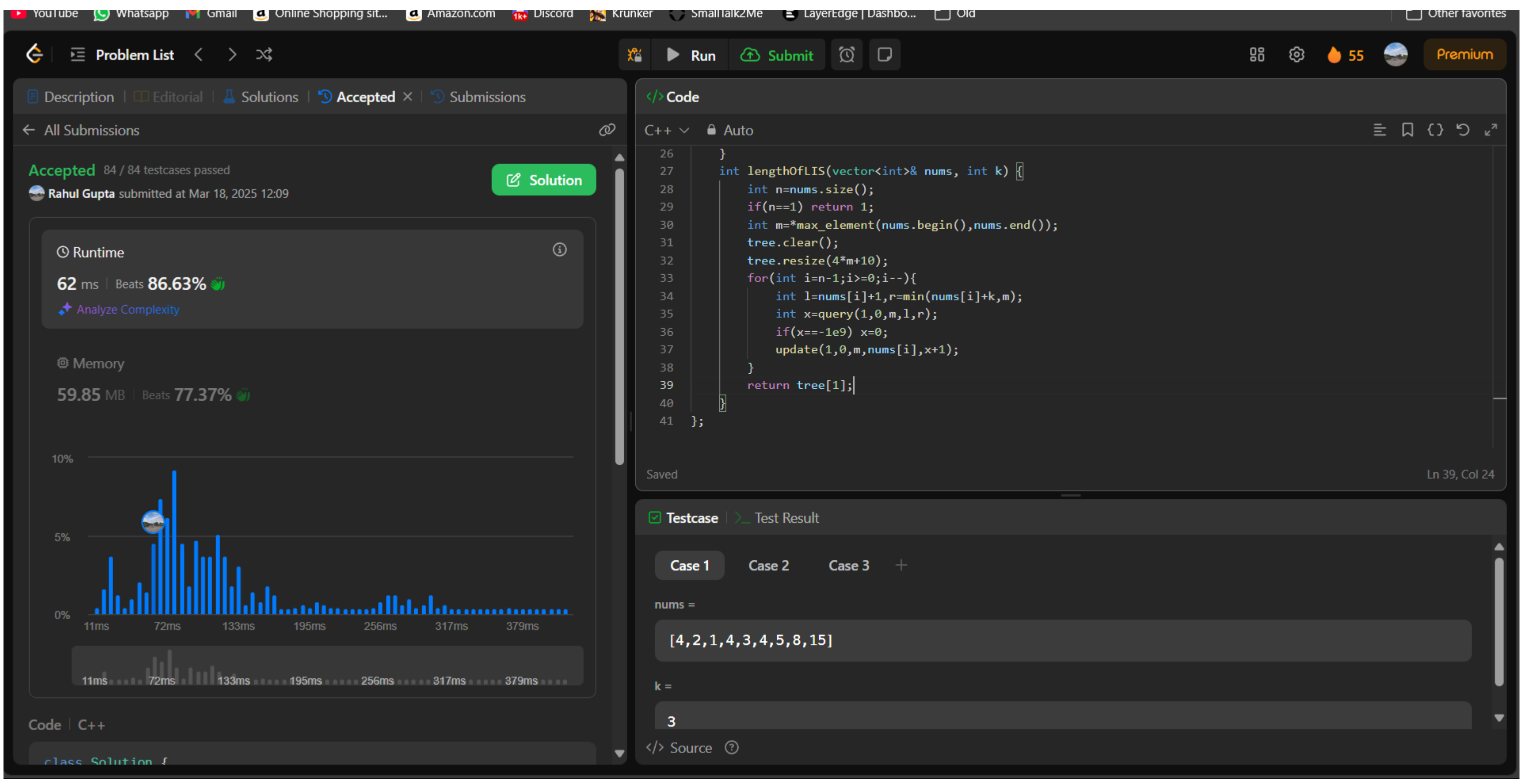
Ln 14, Col 14

nums =

```
[1,3,2,3,1]
```

Q10.) Longest Increasing Subsequence II

```
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;
        int left=query(2*node,st,mid,x,y);
        int right=query(2*node+1,mid+1,end,x,y);
        return max(left,right);
    }
    int lengthOfLIS(vector<int>& nums, int k) {
        int n=nums.size();
        if(n==1) return 1;
        int m=*max_element(nums.begin(),nums.end());
        tree.clear();
        tree.resize(4*m+10);
        for(int i=n-1;i>=0;i--){
            int l=nums[i]+1,r=min(nums[i]+k,m);
            int x=query(1,0,m,l,r);
            if(x==-1e9) x=0;
            update(1,0,m,nums[i],x+1);
        }
        return tree[1];
    }
};
```



Accepted 84 / 84 testcases passed

Rahul Gupta submitted at Mar 18, 2025 12:09

Runtime

62 ms | Beats 86.63%

Analyze Complexity

Memory

59.85 MB | Beats 77.37%

Code | C++

```
class Solution {
```

```
    }
    int lengthOfLIS(vector<int>& nums, int k) {
        int n=nums.size();
        if(n==1) return 1;
        int m=*max_element(nums.begin(),nums.end());
        tree.clear();
        tree.resize(4*m+10);
        for(int i=n-1;i>=0;i--){
            int l=nums[i]+1,r=min(nums[i]+k,m);
            int x=query(1,0,m,l,r);
            if(x==-1e9) x=0;
            update(1,0,m,nums[i],x+1);
        }
        return tree[1];
    }
};
```

Saved

Ln 39, Col 24

nums =

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

k =

```
3
```