AP Assignment

Student Name: Shreya Singh **UID:** 22BCS12423
Branch: BE-CSE **Section/Group:** 606-B
Semester: 6th **Date :** 18/03/25
Subject Name: AP Lab-2 **Subject Code:** 22CSP-351

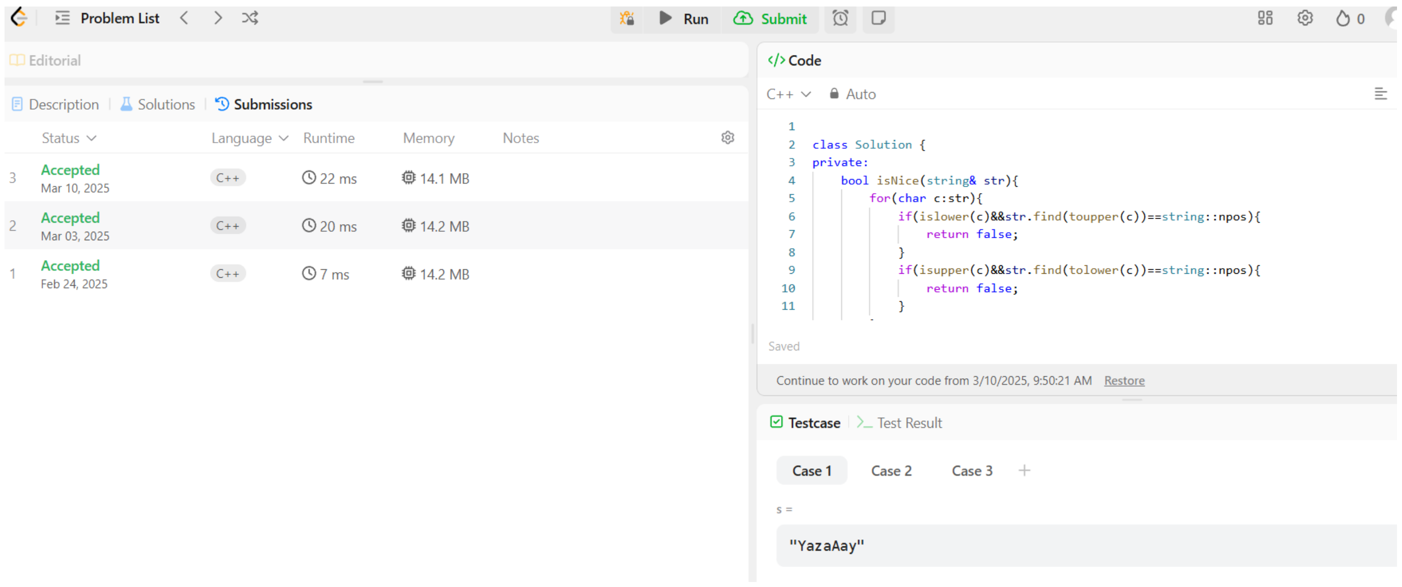
Problem 1

1. **Aim:** Longest Nice Substring

2. **Implementation/Code:**

```
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;
    }
};
```

3. Output:



	Status	Language	Runtime	Memory	Notes
3	Accepted Mar 10, 2025	C++	22 ms	14.1 MB	
2	Accepted Mar 03, 2025	C++	20 ms	14.2 MB	
1	Accepted Feb 24, 2025	C++	7 ms	14.2 MB	

C++ Auto

```

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;
            }
```

Saved

Continue to work on your code from 3/10/2025, 9:50:21 AM Restore

s =

```
"YazaAay"
```

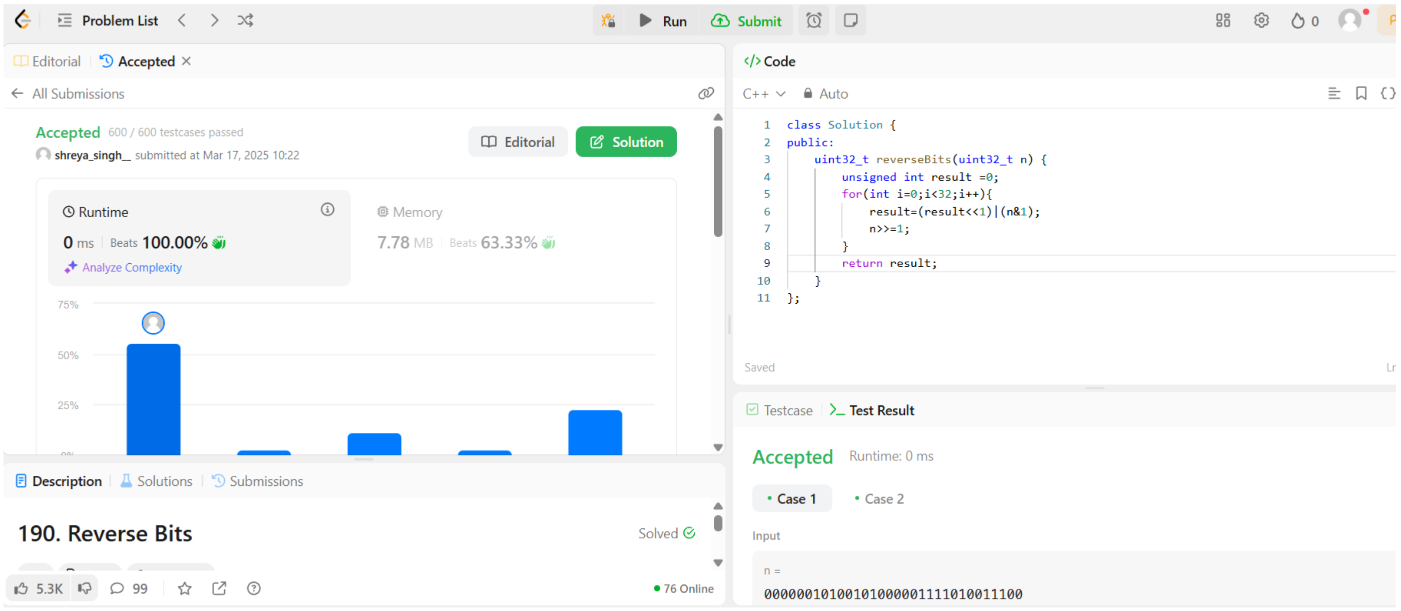
Problem 2

1. Aim: Reverse bits of a given 32 bits unsigned integer.

2. Implementation/Code:

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

3. Output:



Problem List · Run · Submit

Editorial | Accepted ×

All Submissions

Accepted 600 / 600 testcases passed
shreya_singh__ submitted at Mar 17, 2025 10:22

Runtime: 0 ms | Beats 100.00%
Memory: 7.78 MB | Beats 63.33%

190. Reverse Bits — Solved

Code — C++ — Auto

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

Saved

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 · Case 2

Input

```
n =
00000010100101000001111010011100
```

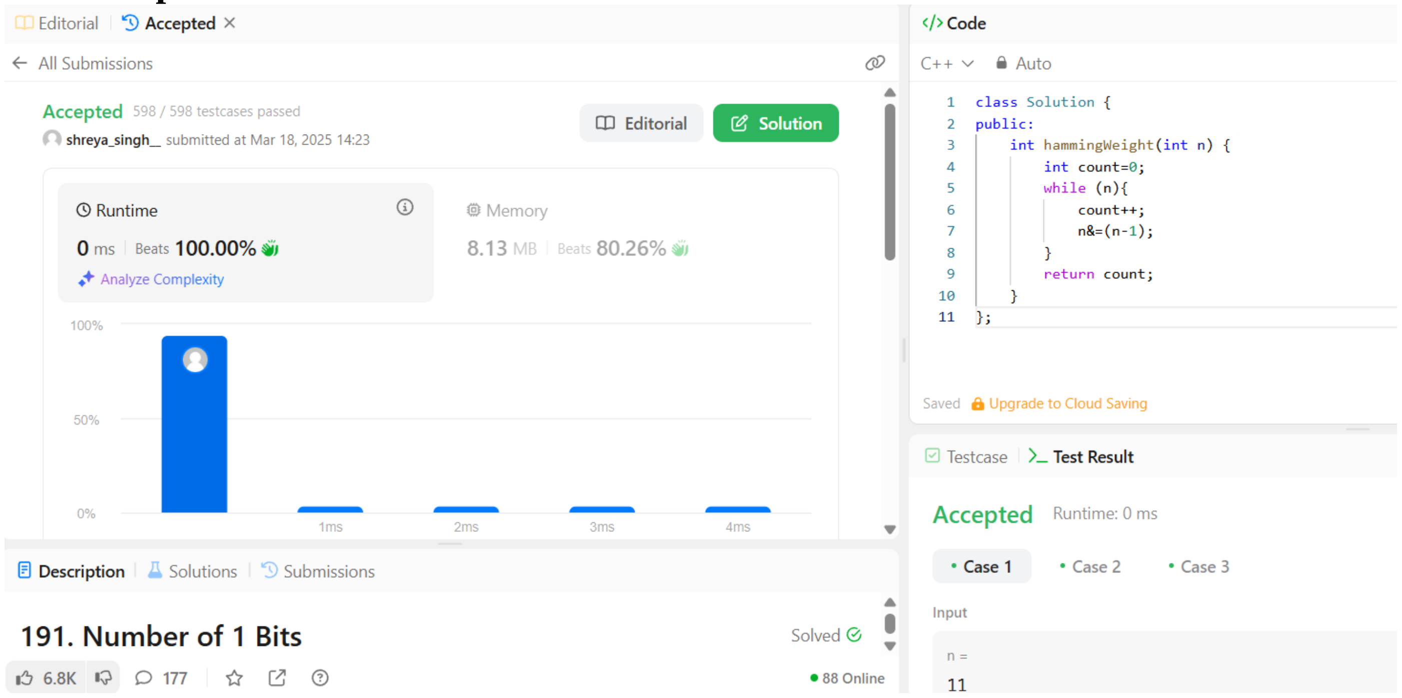
Problem 3

1. **Aim:** Number of 1 Bits.

2. **Implementation/Code:**

```
class Solution {
public:
int hammingWeight(int n) {
        int count=0;
        while (n){
           count++;
           n&=(n-1);
        }
        return count;
    }
};
```

3. **Output:**



Accepted 598 / 598 testcases passed
shreya_singh__ submitted at Mar 18, 2025 14:23

Runtime
0 ms | Beats 100.00%

Memory
8.13 MB | Beats 80.26%

191. Number of 1 Bits
Solved
6.8K 177
88 Online

```
class Solution {
public:
    int hammingWeight(int n) {
        int count=0;
        while (n){
            count++;
            n&=(n-1);
        }
        return count;
    }
};
```

Accepted Runtime: 0 ms

n =
11

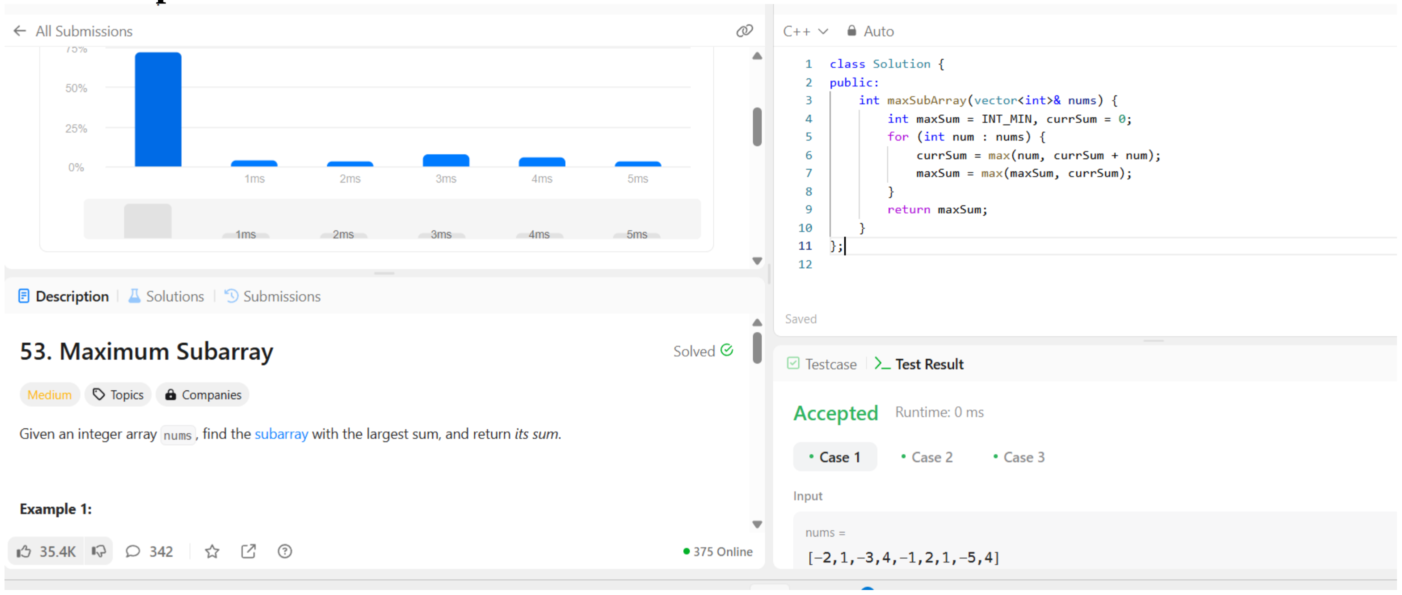
Problem 4

1. **Aim:** Maximum Subarray

2. **Implementation/Code:**

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int maxSum = INT_MIN, currSum = 0;
        for (int num : nums) {
            currSum = max(num, currSum + num);
            maxSum = max(maxSum, currSum);
        }
        return maxSum;
    }
};
```

3. **Output:**



```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int maxSum = INT_MIN, currSum = 0;
        for (int num : nums) {
            currSum = max(num, currSum + num);
            maxSum = max(maxSum, currSum);
        }
        return maxSum;
    }
};
```

53. Maximum Subarray

Given an integer array `nums`, find the subarray with the largest sum, and return *its sum*.

Example 1:

Accepted Runtime: 0 ms

Input

nums =
[-2,1,-3,4,-1,2,1,-5,4]

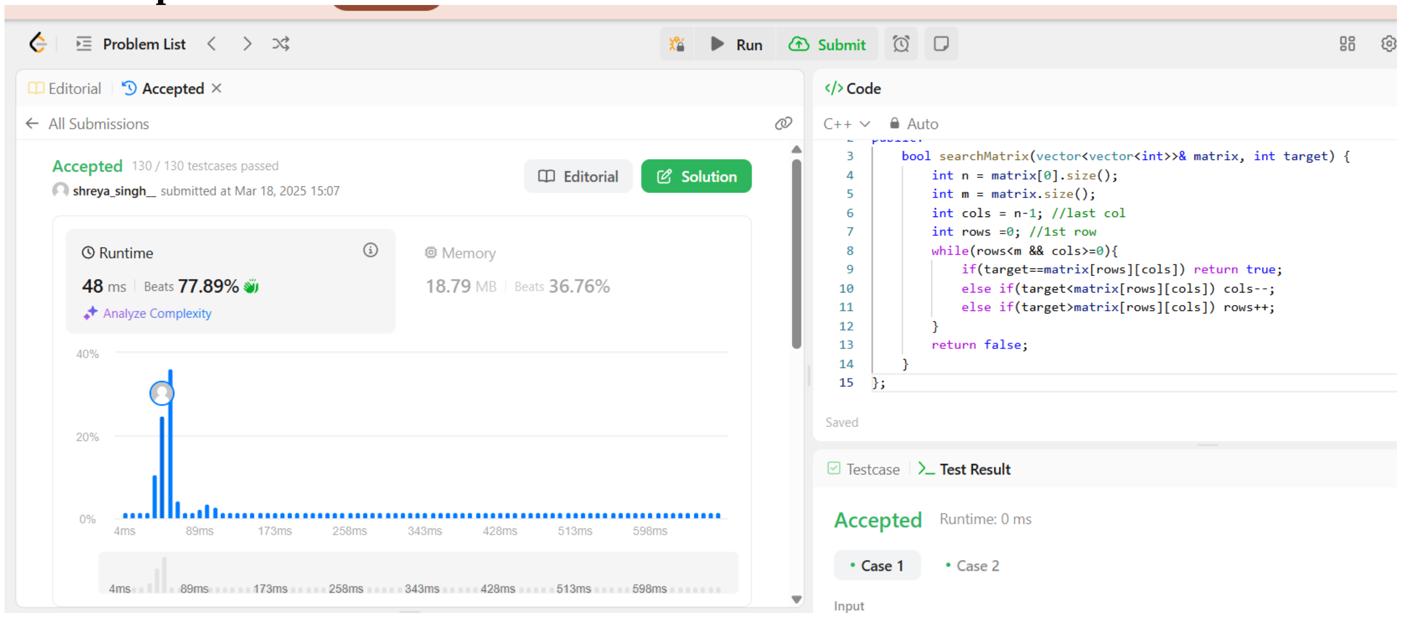
Problem 5

1. **Aim:** Search a 2D Matrix II

2. **Implementation/Code:**

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int n = matrix[0].size();
        int m = matrix.size();
        int cols = n-1; //last col
        int rows =0; //1st row
        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;
    }
};
```

3. **Output:**



Accepted 130 / 130 testcases passed
shreya_singh__ submitted at Mar 18, 2025 15:07

Runtime
48 ms | Beats 77.89%

Memory
18.79 MB | Beats 36.76%

```
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int n = matrix[0].size();
        int m = matrix.size();
        int cols = n-1; //last col
        int rows =0; //1st row
        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 Runtime: 0 ms

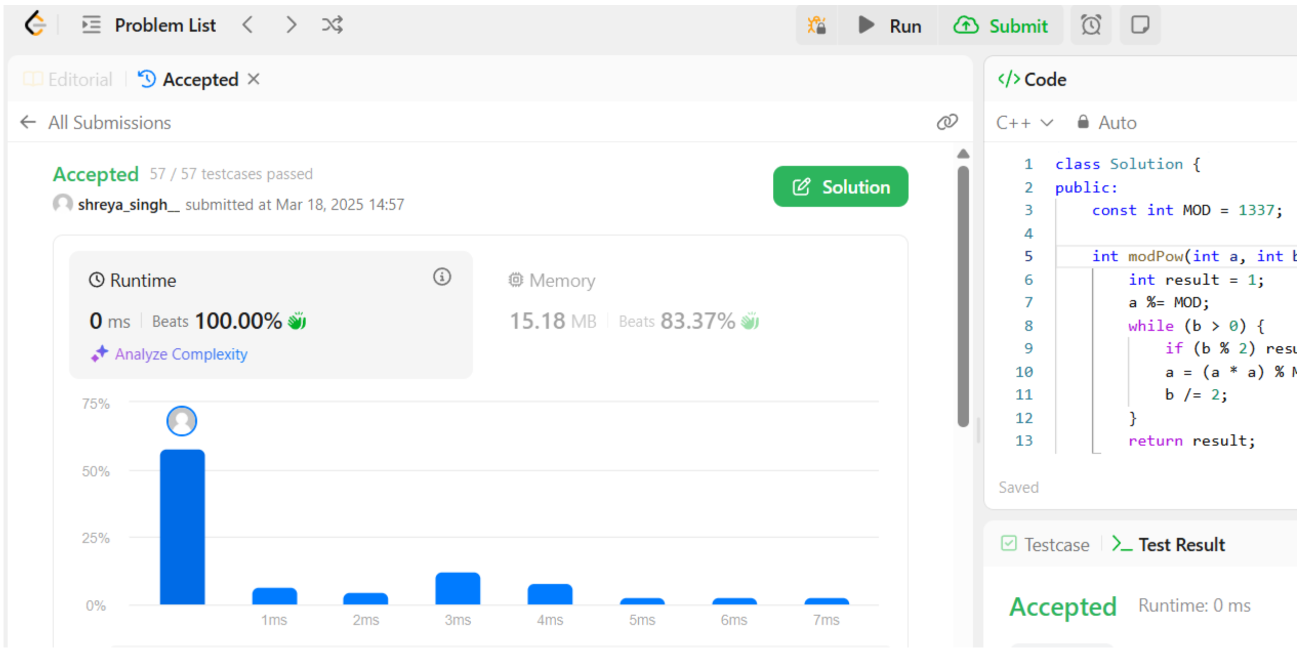
Problem 6

1. **Aim:** Super Pow.

2. **Implementation/Code:**

```
class Solution {
public:
   const int MOD = 1337;
    int modPow(int a, int b) {
        int result = 1;
        a %= MOD;
        while (b > 0) {
            if (b % 2) result = (result * a) % MOD;
            a = (a * a) % MOD;
            b /= 2;
        }
        return result;
    }
    int superPow(int a, vector<int>& b) {
        int result = 1;
        for (int digit : b) {
            result = modPow(result, 10) * modPow(a, digit) % MOD;
        }
        return result;
    }
};
```

3. **Output:**



Accepted 57 / 57 testcases passed
shreya_singh__ submitted at Mar 18, 2025 14:57

Runtime
0 ms | Beats 100.00%

Memory
15.18 MB | Beats 83.37%

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

    int modPow(int a, int b
        int result = 1;
        a %= MOD;
        while (b > 0) {
            if (b % 2) resu
            a = (a * a) % M
            b /= 2;
        }
        return result;
```

Accepted Runtime: 0 ms

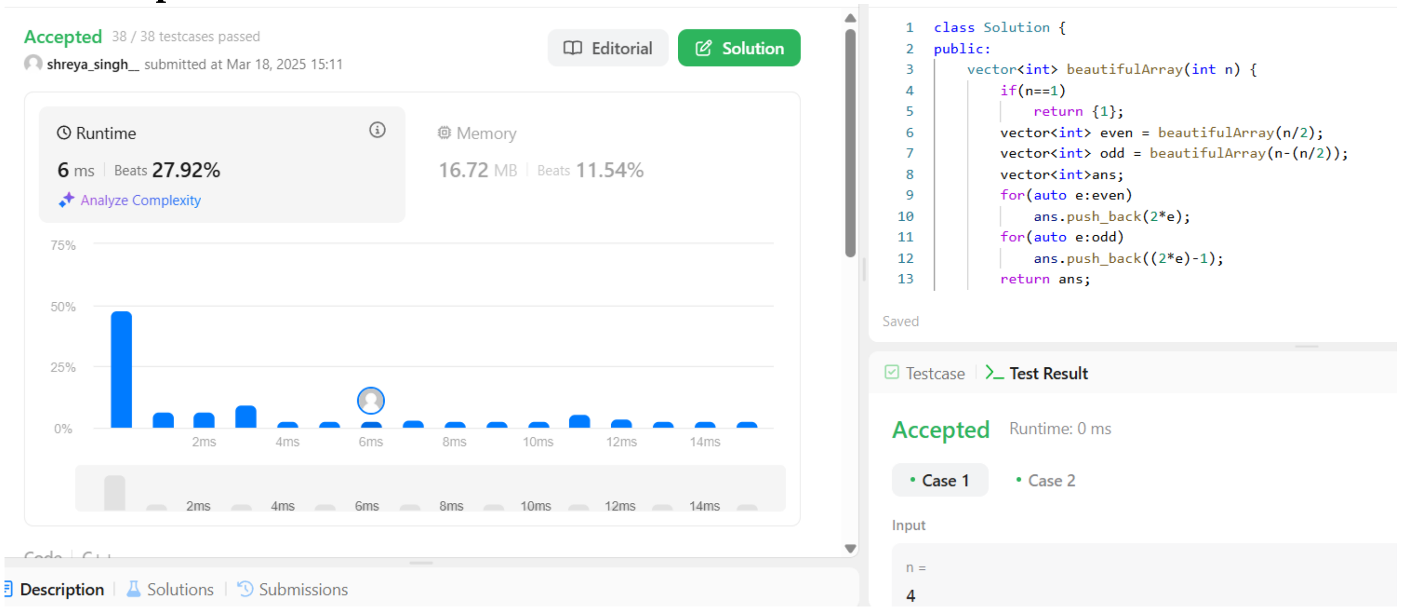
Problem 7

1. **Aim:** Beautiful Array.

2. **Implementation/Code:**

```
class Solution {
public:
    vector<int> beautifulArray(int n) {
        if(n==1)
            return {1};
        vector<int> even = beautifulArray(n/2);
        vector<int> odd = beautifulArray(n-(n/2));
        vector<int>ans;
        for(auto e:even)
            ans.push_back(2*e);
        for(auto e:odd)
            ans.push_back((2*e)-1);
        return ans;
    }
};
```

3. **Output:**



Accepted 38 / 38 testcases passed
shreya_singh__ submitted at Mar 18, 2025 15:11

Runtime
6 ms | Beats 27.92%

Memory
16.72 MB | Beats 11.54%

```
class Solution {
public:
    vector<int> beautifulArray(int n) {
        if(n==1)
            return {1};
        vector<int> even = beautifulArray(n/2);
        vector<int> odd = beautifulArray(n-(n/2));
        vector<int>ans;
        for(auto e:even)
            ans.push_back(2*e);
        for(auto e:odd)
            ans.push_back((2*e)-1);
        return ans;
```

Accepted Runtime: 0 ms

Input

n =
4

Problem 8

1. **Aim:** The Skyline Problem.

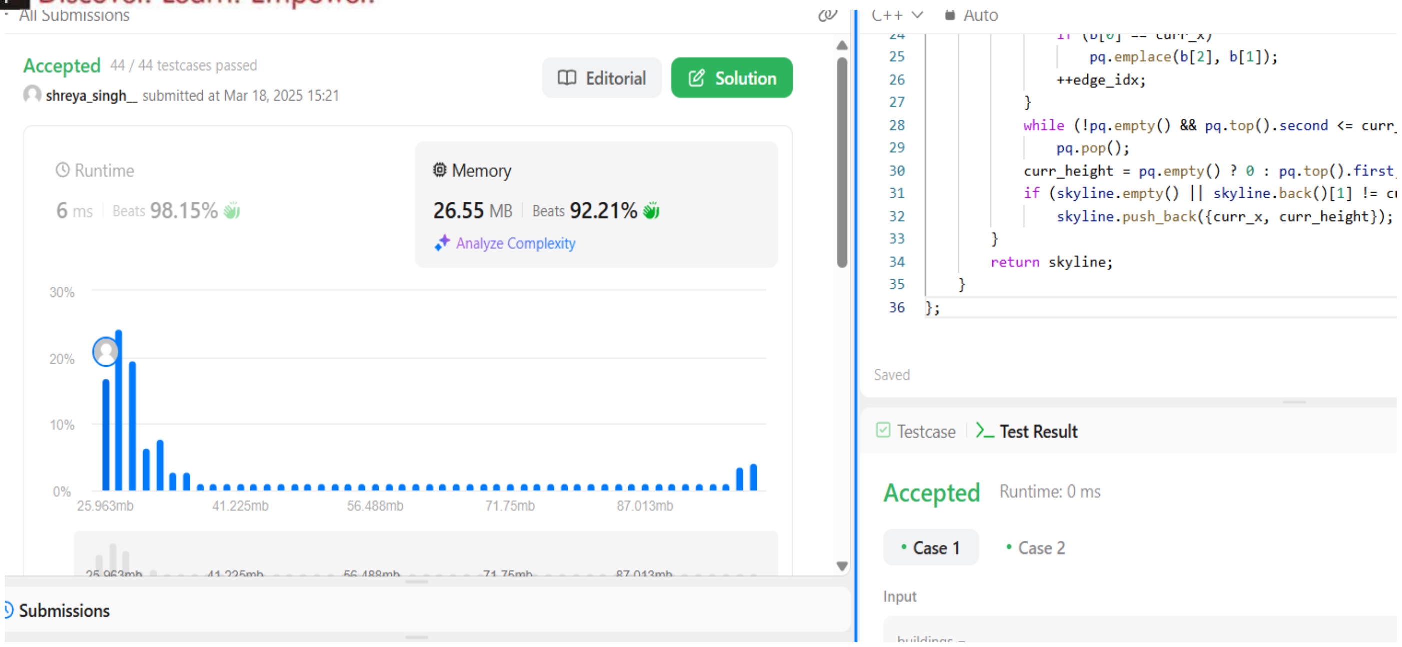
2. **Implementation/Code:**

```
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)
                skyline.push_back({curr_x, curr_height});
        }
        return skyline;
    }
};
```

3. **Output:**



All Submissions

Accepted 44 / 44 testcases passed

shreya_singh__ submitted at Mar 18, 2025 15:21

Editorial Solution

Runtime
6 ms | Beats 98.15%

Memory
26.55 MB | Beats 92.21%

Analyze Complexity

Submissions

C++ Auto

```
                if (b[0] == curr_x)
                    pq.emplace(b[2], b[1]);
                    ++edge_idx;
                }
                while (!pq.empty() && pq.top().second <= curr_
                    pq.pop();
                curr_height = pq.empty() ? 0 : pq.top().first
                if (skyline.empty() || skyline.back()[1] != c
                    skyline.push_back({curr_x, curr_height});
            }
            return skyline;
        }
};
```

Saved

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

buildings =

Problem 9

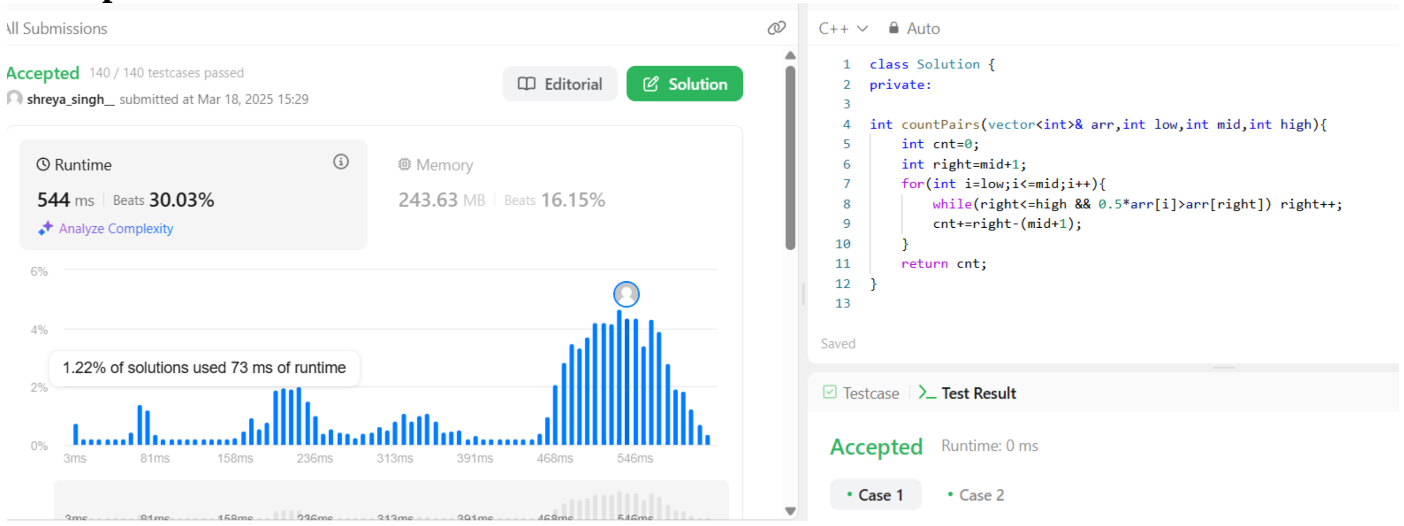
1. **Aim:** Reverse Pairs.

2. **Implementation/Code:**

```
   class Solution {
private:
int countPairs(vector<int>& arr,int low,int mid,int high){
        int cnt=0;
        int right=mid+1;
        for(int i=low;i<=mid;i++){
                while(right<=high && 0.5*arr[i]>arr[right]) right++;
                cnt+=right-(mid+1);
        }
        return cnt;
}
void merge(vector<int>& arr,int low,int mid,int high){
        int left=low;
        int right=mid+1;
        vector<int> temp;

while(left<=mid && right<=high){
                if(arr[left]<=arr[right]){
                        temp.push_back(arr[left]);
                        left++;
                }
                else{
                         temp.push_back(arr[right]);
                          right++; }
        }
        while(left<=mid){
                temp.push_back(arr[left]);
                left++;
        }
        while(right<=high){
                temp.push_back(arr[right]);
                right++;
        }
        for(int i=low;i<=high;i++){
                arr[i]=temp[i-low];
        }
}
int mergesort(vector<int>& arr,int low,int high){
        int cnt=0;
        if(low>=high) return cnt;
        int mid=(low+high)/2;
        cnt+=mergesort(arr,low,mid);
        cnt+=mergesort(arr,mid+1,high);
        cnt+=countPairs(arr,low,mid,high);
        merge(arr,low,mid,high);
        return cnt;
}
public:
    int reversePairs(vector<int>& nums) {
        return mergesort(nums,0,nums.size()-1);
    }
};
```

3. Output:



Accepted 140 / 140 testcases passed
shreya_singh__ submitted at Mar 18, 2025 15:29

Runtime
544 ms | Beats 30.03%

Memory
243.63 MB | Beats 16.15%

1.22% of solutions used 73 ms of runtime

```
class Solution {
private:

int countPairs(vector<int>& arr,int low,int mid,int high){
    int cnt=0;
    int right=mid+1;
    for(int i=low;i<=mid;i++){
        while(right<=high && 0.5*arr[i]>arr[right]) right++;
        cnt+=right-(mid+1);
    }
    return cnt;
}

```

Accepted Runtime: 0 ms

Problem 10

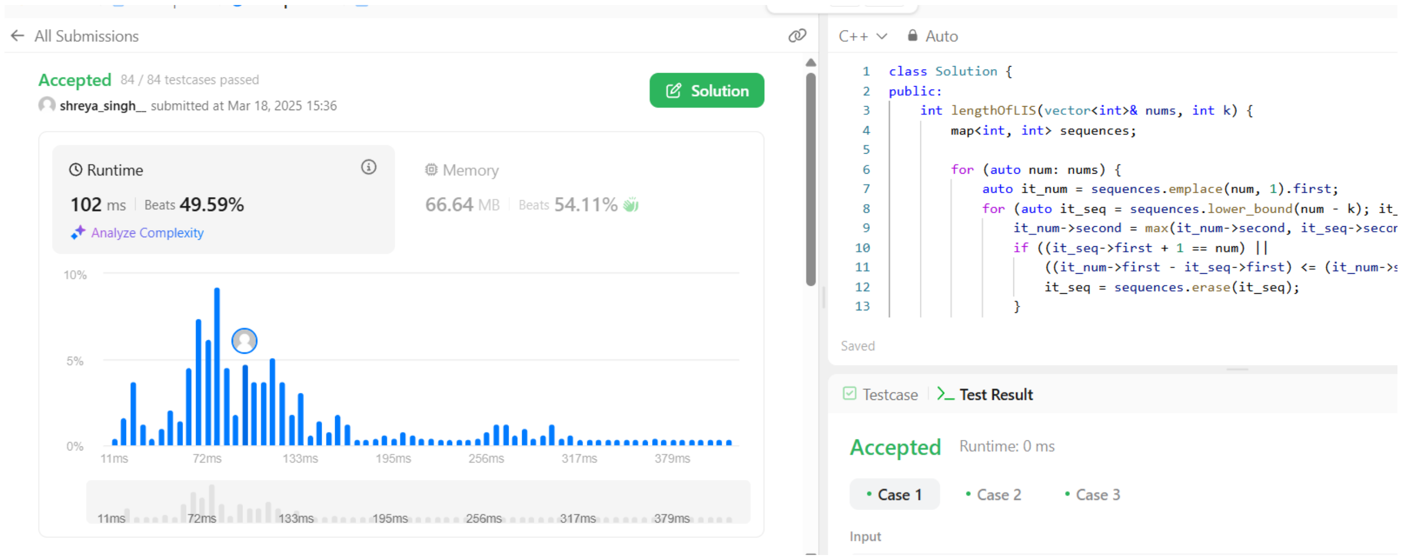
4. Aim: Longest Increasing Subsequence II

5. Implementation/Code:

```
  class Solution {
public:
    int lengthOfLIS(vector<int>& nums, int k) {
        map<int, int> sequences;

        for (auto num: nums) {
            auto it_num = sequences.emplace(num, 1).first;
            for (auto it_seq = sequences.lower_bound(num - k); it_seq != it_num; ) {
                it_num->second = max(it_num->second, it_seq->second + 1);
                if ((it_seq->first + 1 == num) ||
                    ((it_num->first - it_seq->first) <= (it_num->second - it_seq->second))) {
                    it_seq = sequences.erase(it_seq);
                }
                else {
                    ++it_seq;
                }
            }
        }
        return max_element(sequences.begin(), sequences.end(), [](auto s1, auto s2) { return s1.second
< s2.second; })->second;
    }
};
```

6. Output:



Accepted 84 / 84 testcases passed
shreya_singh_ submitted at Mar 18, 2025 15:36

Runtime 102 ms Beats 49.59%

Memory 66.64 MB Beats 54.11%

```
class Solution {
public:
    int lengthOfLIS(vector<int>& nums, int k) {
        map<int, int> sequences;

        for (auto num: nums) {
            auto it_num = sequences.emplace(num, 1).first;
            for (auto it_seq = sequences.lower_bound(num - k); it_
                it_num->second = max(it_num->second, it_seq->secon
                if ((it_seq->first + 1 == num) ||
                    ((it_num->first - it_seq->first) <= (it_num->s
                    it_seq = sequences.erase(it_seq);
                }
```

Accepted Runtime: 0 ms