**Name : Anjali**

**UID : 22BCS10665  
Batch : 22BCS\_IOT-614(B)**

**Assignment – 7**

**Problem-1 : House Robber**  
**Code:**

#include <vector>

using namespace std;

class Solution {

public:

int rob(vector<int>& nums) {

int n = nums.size();

if (n == 0) return 0;

if (n == 1) return nums[0];

int prev2 = 0, prev1 = 0;

for (int num : nums) {

int temp = max(prev1, prev2 + num);

prev2 = prev1;

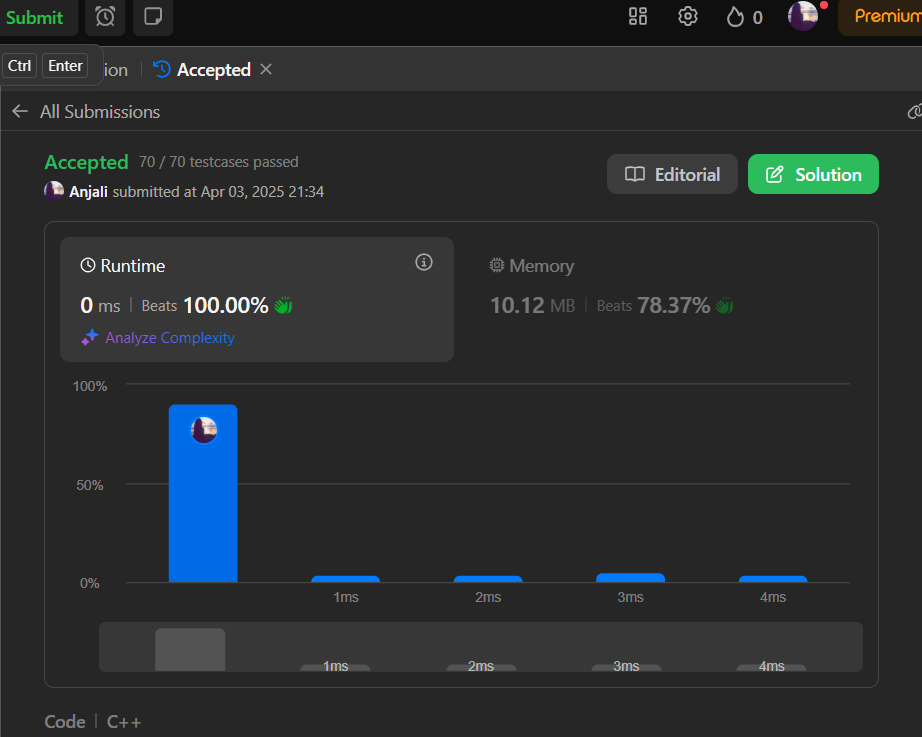
prev1 = temp;

}

return prev1;

}

};

**Output SS:**

**Problem-2 : Jump Game**  
**Code:**

#include <vector>

using namespace std;

class Solution {

public:

    bool canJump(vector<int>& nums) {

        int maxReach = 0;

        int n = nums.size();

        for (int i = 0; i < n; ++i) {

            if (i > maxReach) {

                return false;

            }

            maxReach = max(maxReach, i + nums[i]);

            if (maxReach >= n - 1) {

                return true;

            }

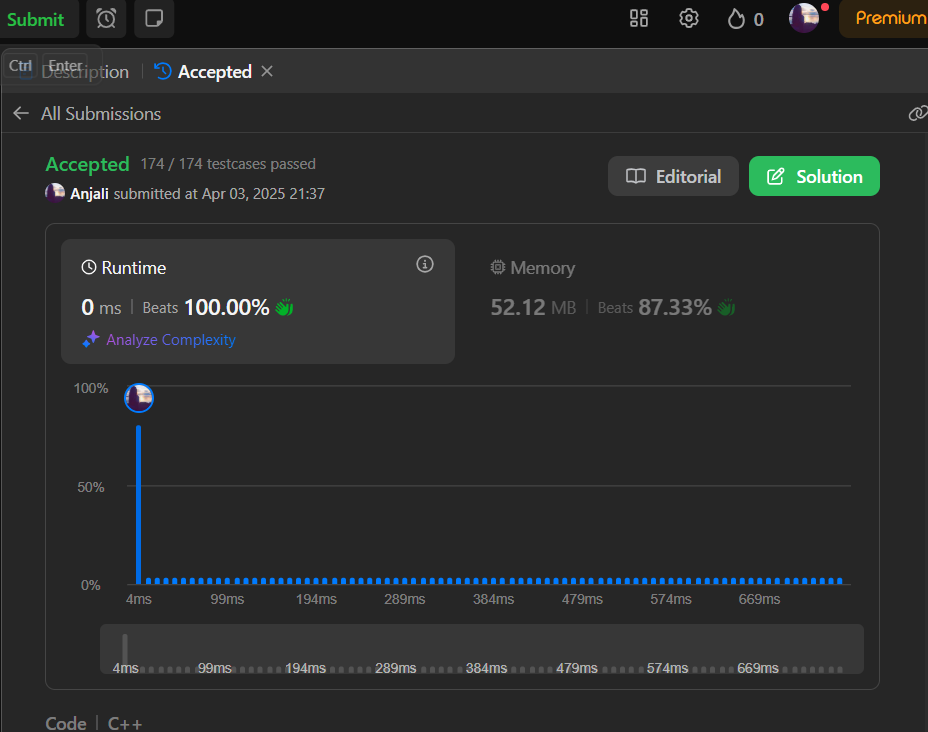
        }

        return false;

    }

};

**Output SS:**



**Problem-3 : Maximum Product Subarray**  
**Code:**

#include <vector>

#include <algorithm>

using namespace std;

class Solution {

public:

int maxProduct(vector<int>& nums) {

int n = nums.size();

int maxProd = nums[0], minProd = nums[0], result = nums[0];

for (int i = 1; i < n; i++) {

if (nums[i] < 0)

swap(maxProd, minProd);

maxProd = max(nums[i], maxProd \* nums[i]);

minProd = min(nums[i], minProd \* nums[i]);

result = max(result, maxProd);

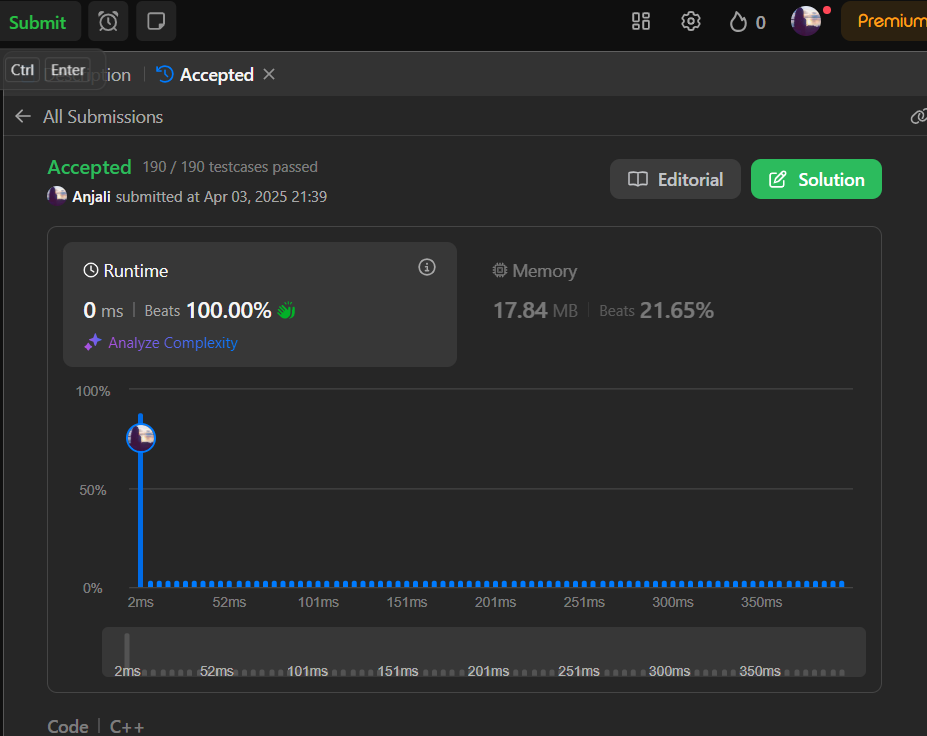
}

return result;

}

};

**Output SS:**



**Problem-4 : Perfect Squares**  
**Code:**

#include <vector>

#include <cmath>

using namespace std;

class Solution {

public:

int numSquares(int n) {

vector<int> dp(n + 1, INT\_MAX);

dp[0] = 0;

for (int i = 1; i <= n; i++) {

for (int j = 1; j \* j <= i; j++) {

dp[i] = min(dp[i], dp[i - j \* j] + 1);

}

}

return dp[n];

}

};

**Output SS:**

