Advanced Pragramming

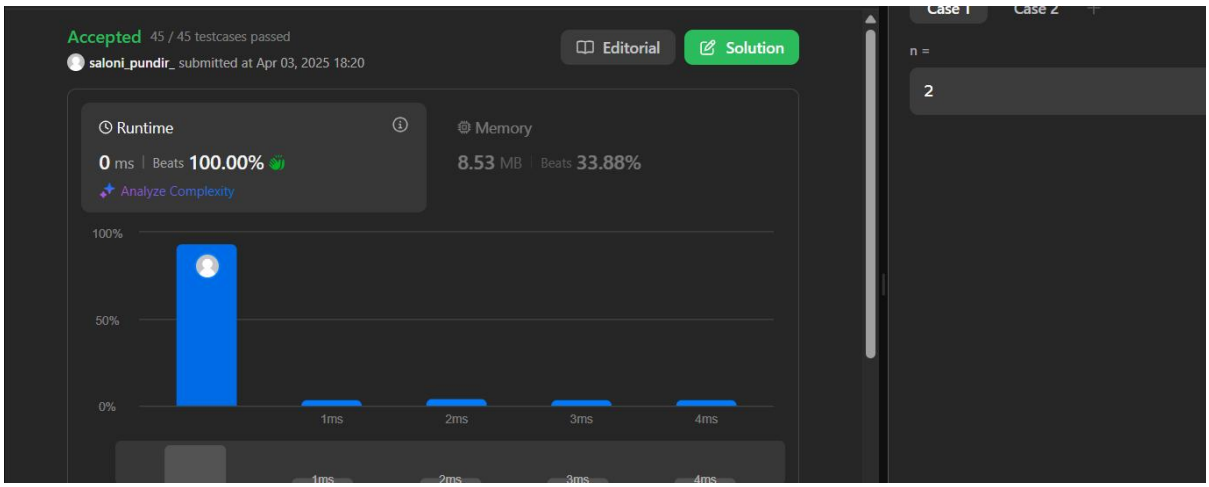
ASSIGNMENT 07

Q1. Climbing Stairs

Code:

```cpp
class Solution {
public:
    int solve(int n,vector<int> &dp){
        //base case
        if(n<=2)
          return n;

        if(dp[n]!=-1)
          return dp[n];

        dp[n]=solve(n-1,dp)+solve(n-2,dp);
        return dp[n];
    }
    int climbStairs(int n) {
        if(n<=2)
         return n;
        vector<int> dp(n+1,-1);
        return solve(n,dp);
    }
};
```

Output:

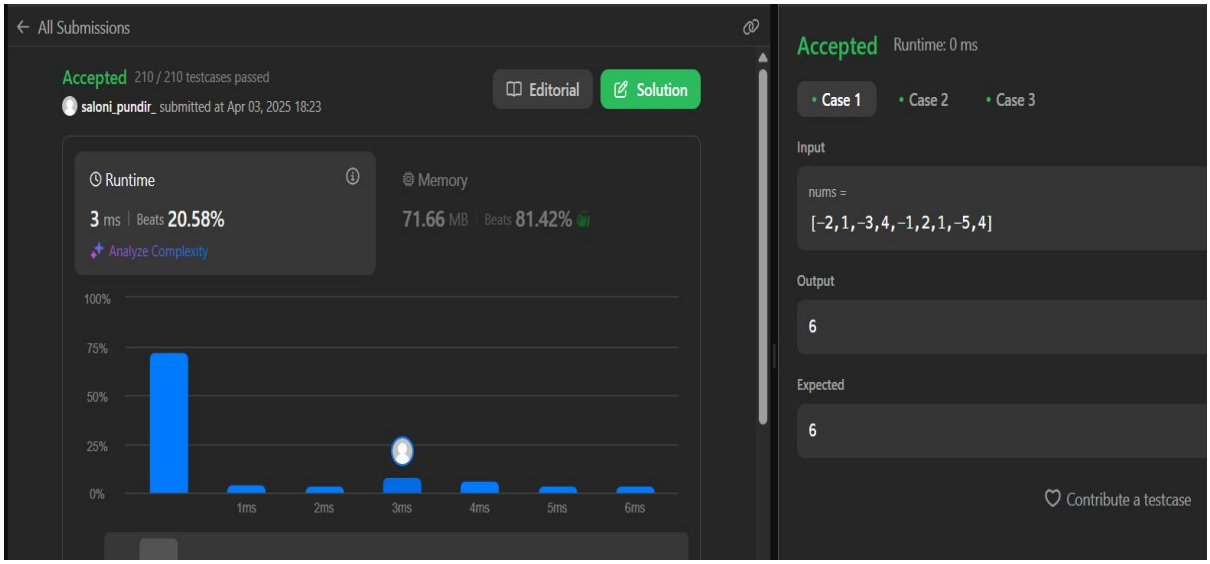


Q2. Maximum Subarray

Code:

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

        for (int n : nums) {
            if (total < 0) {
                total = 0;
            }

            total += n;
            res = max(res, total);
        }

        return res;
    }
};
```

Output:



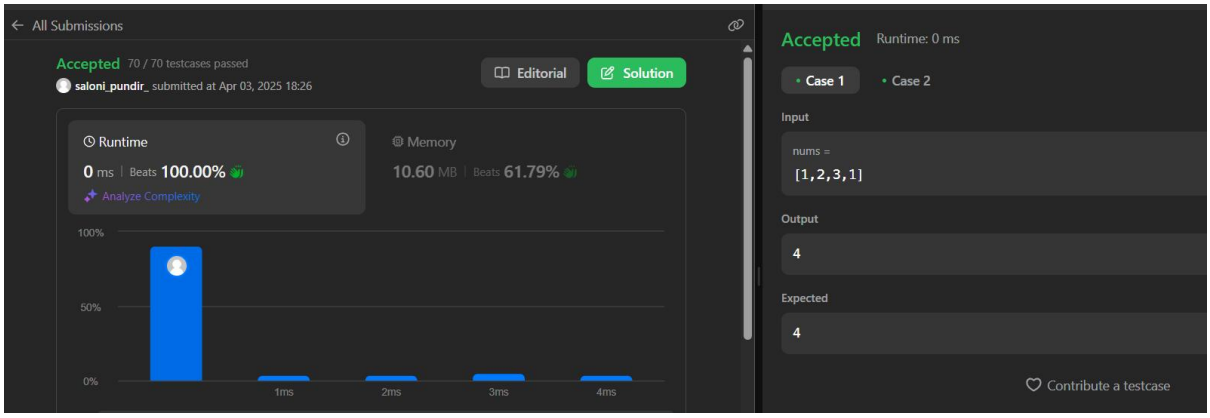
Q3. House Robber.

Code:

C++ Auto

```cpp
class Solution {
public:
    int rob(vector<int>& nums) {
        int n = nums.size();

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

        vector<int> dp(n, 0);

        dp[0] = nums[0];
        dp[1] = max(nums[0], nums[1]);

        for (int i = 2; i < n; i++) {
            dp[i] = max(dp[i - 1], nums[i] + dp[i - 2]);
        }

        return dp[n - 1];
    }
};
```

Output:



Q4. Jump Game.

Code:

```cpp
class Solution {
public:
    bool canJump(vector<int>& nums) {
        int goal = nums.size() - 1;

        for (int i = nums.size() - 2; i >= 0; i--) {
            if (i + nums[i] >= goal) {
                goal = i;
            }
        }

        return goal == 0;
    }
};
```

Output: