“**Experiment 3.3”**

**Aim:**

To demonstrate the concept of Dynamic Programming.

**Objective:**

• The objective is to build problem solving capability and to learn the basic concepts of data structures.

• The implementation of ‘Best Time to buy and sell the stock’ problem brushes up the concept of greedy approach.

• The implementation of ‘Climbing Stairs’ problem brushes up the concept of greedy approach.

**Problem 1: “Best Time to buy and sell the stock”**

<https://leetcode.com/problems/best-time-to-buy-and-sell-stock/>

You are given an array prices where prices[i] is the price of a given stock on the ith day.

You want to maximize your profit by choosing a single day to buy one stock and choosing a different day in the future to sell that stock.

Return the maximum profit you can achieve from this transaction. If you cannot achieve any profit, return 0.

**Code:**

class Solution {

public:

    int maxProfit(vector<int>& prices) {

        int minPri=INT\_MAX, maxPro=0;

        for(int i=0;i<prices.size();i++){       // O(n)

            minPri=min(minPri,prices[i]);

            maxPro=max(maxPro,prices[i]-minPri);

        }

        return maxPro;

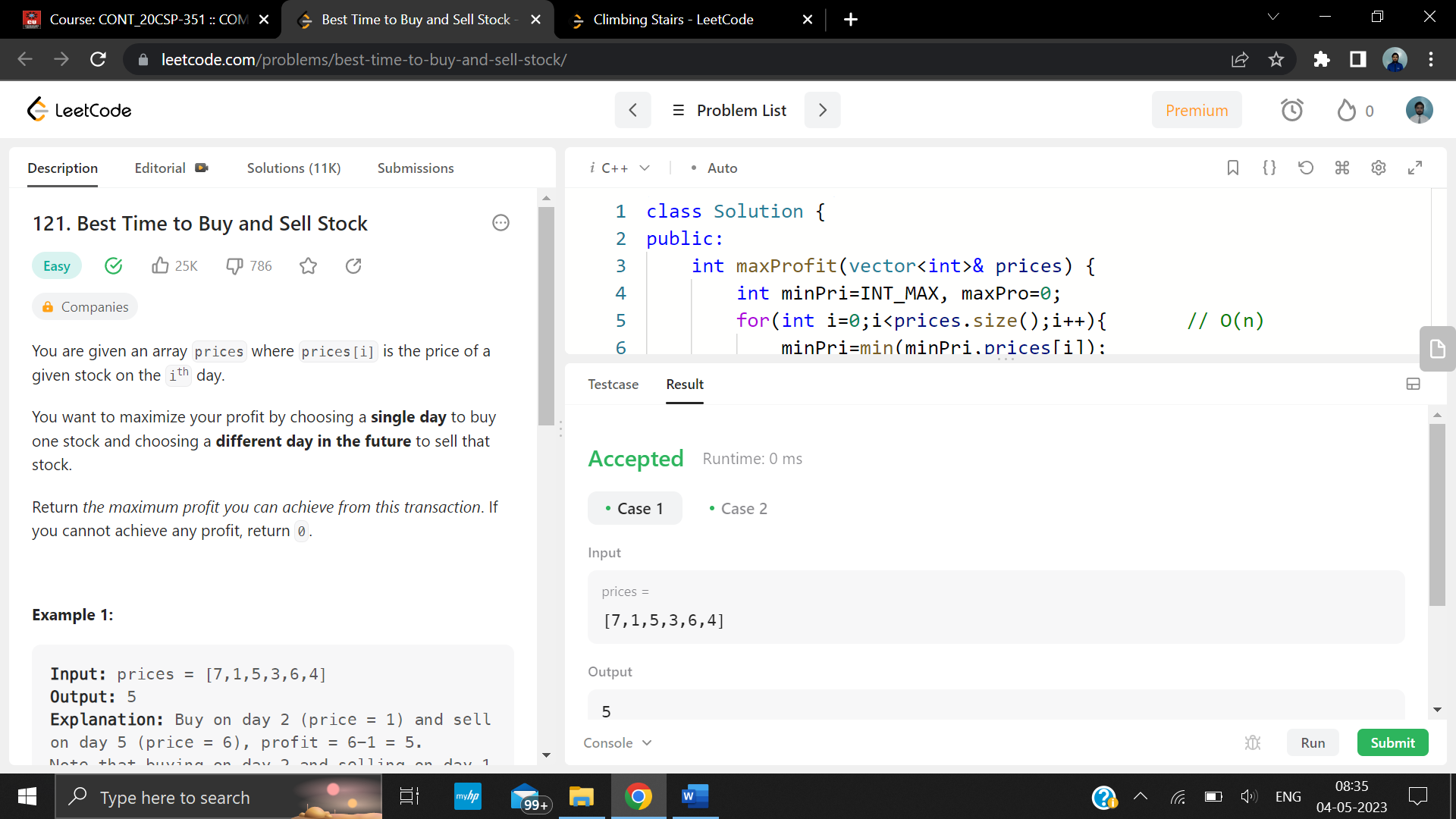
    }

};

// Time Complexity: O(n)

// Space Complexity: O(1)

**Output:**



**Problem 2: “Climbing Stairs”**

<https://leetcode.com/problems/climbing-stairs/description/>

You are climbing a staircase. It takes n steps to reach the top.

Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

**Code:**

class Solution {

public:

    int climbStairs(int n) {

        // base cases

        if(n <= 0) return 0;

        if(n == 1) return 1;

        if(n == 2) return 2;

        int one\_step\_before = 2;

        int two\_steps\_before = 1;

        int all\_ways = 0;

        for(int i=2; i<n; i++){

            all\_ways = one\_step\_before + two\_steps\_before;

            two\_steps\_before = one\_step\_before;

            one\_step\_before = all\_ways;

        }

        return all\_ways;

    }

};

**Output:**

