**Experiment 8.1**

**Student Name:** Sahil Kaundal **UID:** 21BCS8197

**Branch:** BE CSE (Lateral Entry) **Section/Group:** 616/A

**Semester:** 6th **Date of Performance:** 01/05/2023

**Subject Name:** CC-2 Lab **Subject Code:** 20CSP-351

1. **Aim/Overview of the practical:**

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.

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

1. **Apparatus / Simulator Used:**

* Windows 7 or above
* Google Chrome

1. **Objective:**

* To understand the concept of Greedy.

A greedy algorithm is an approach for solving a problem by selecting the best option available at the moment. It doesn't worry whether the current best result will bring the overall optimal result. The algorithm never reverses the earlier decision even if the choice is wrong. It works in a top-down approach.

1. **Code:**

class Solution {

    public int maxProfit(int[] prices) {

        //In constraints it is given that

        //0 <= prices[i] <= 104

        int min = 10000;

        //Profit will be 0, if no transaction are done.

        int maxDiff = 0;

        int size = prices.length;

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

            //We need to find Min value

            min = Math.min(prices[i], min);

            //We need to find maxProfit which is Difference between

            //currentPrice - min, then compare with maxDiff

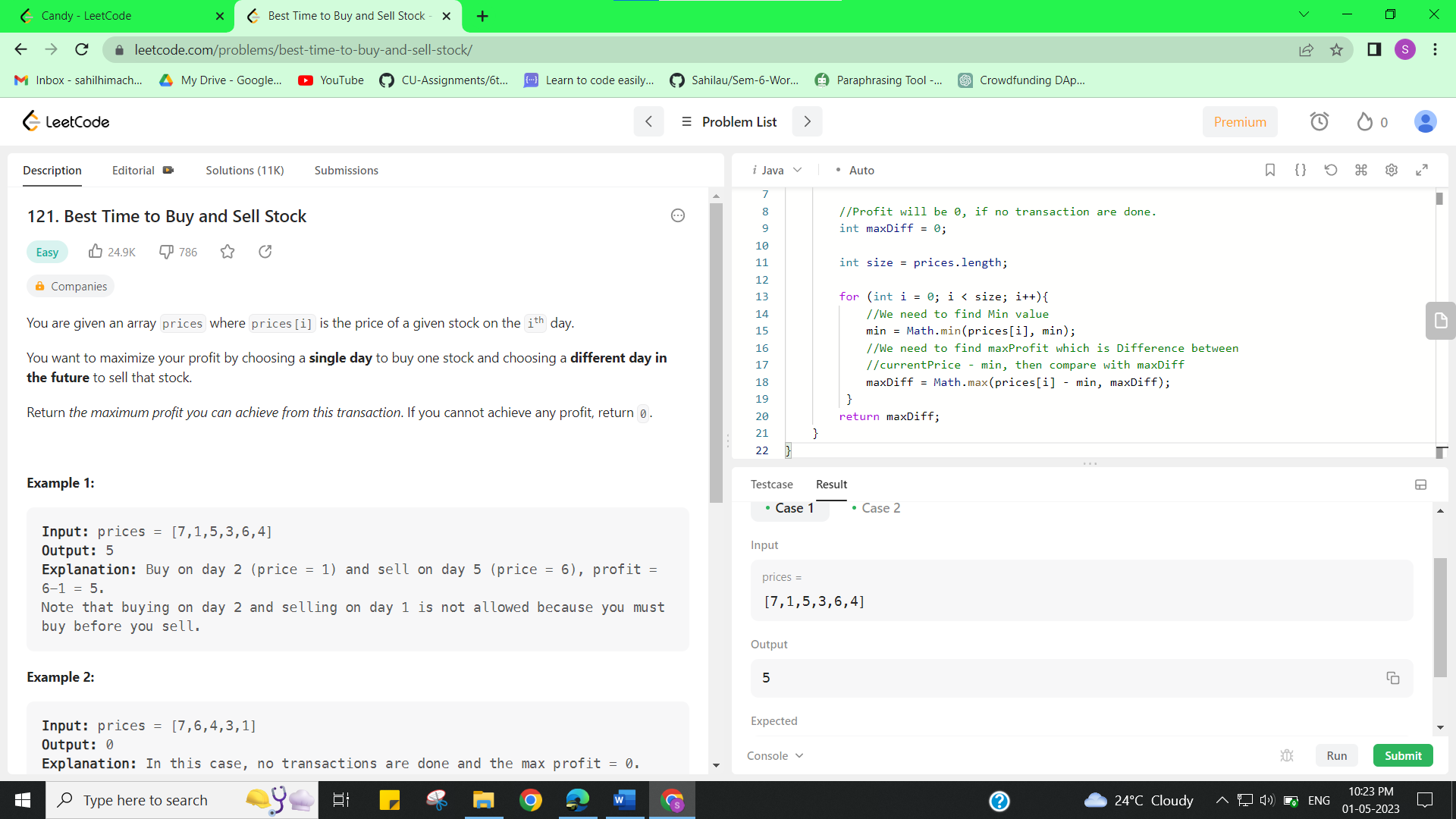
            maxDiff = Math.max(prices[i] - min, maxDiff);

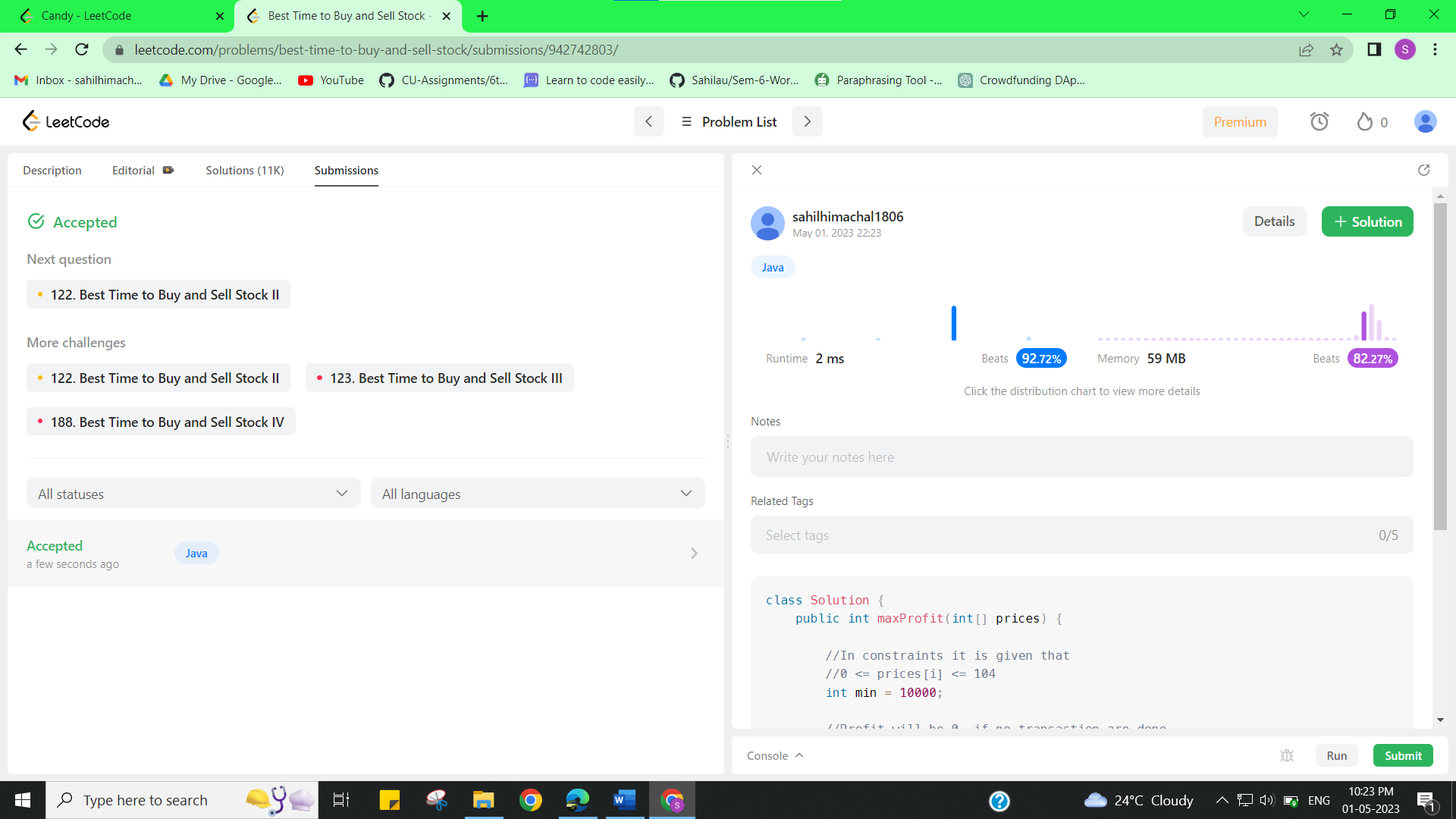
         }

        return maxDiff;

    }}

**4. Result/Output/Writing Summary:**





**Experiment 8.2**

1. **Aim/Overview of the practical:**

Candy

There are n children standing in a line. Each child is assigned a rating value given in the integer array ratings.

You are giving candies to these children subjected to the following requirements:

* Each child must have at least one candy.
* Children with a higher rating get more candies than their neighbors.

Return *the minimum number of candies you need to have to distribute the candies to the children*.

<https://leetcode.com/problems/candy/>

1. **Apparatus / Simulator Used:**

* Windows 7 or above
* Google Chrome

1. **Objective:**

* To understand the concept of Greedy.

A greedy algorithm is an approach for solving a problem by selecting the best option available at the moment. It doesn't worry whether the current best result will bring the overall optimal result. The algorithm never reverses the earlier decision even if the choice is wrong. It works in a top-down approach.

1. **Code:**

class Solution {

        public int candy(int[] nums) {

            int ans = 0;

            int n = nums.length;

            int[] candy = new int[n];

            for(int i=nums.length-1;i>0;i--){

                if(nums[i-1]>nums[i]){

                    candy[i-1] = candy[i]+1;

                }

            }

            for(int i=0;i<nums.length-1;i++)

            {

                if(nums[i]<nums[i+1] && candy[i]>=candy[i+1])

                {

                    candy[i+1] = candy[i]+1;

                }

                ans+=candy[i];

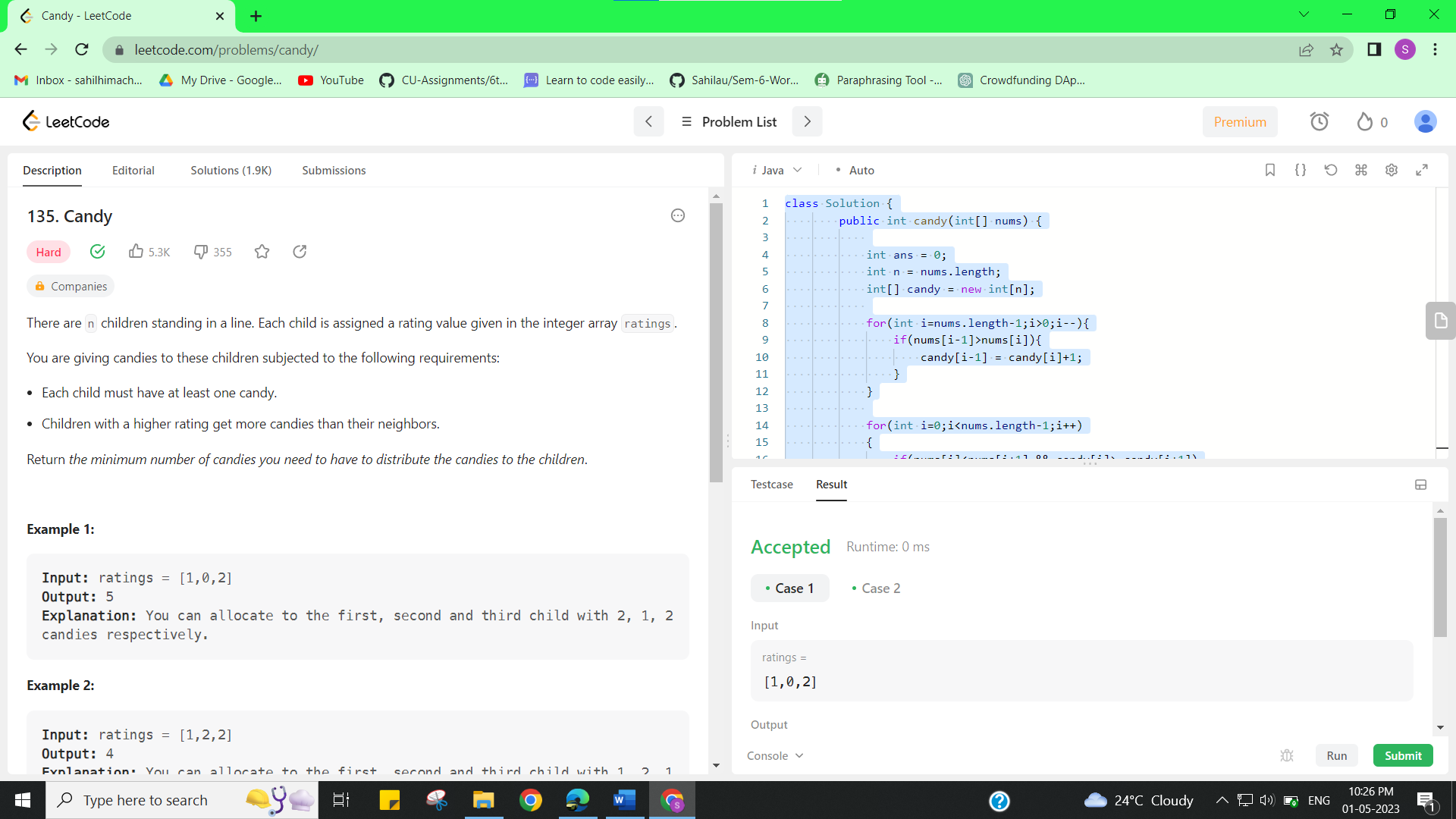
            }

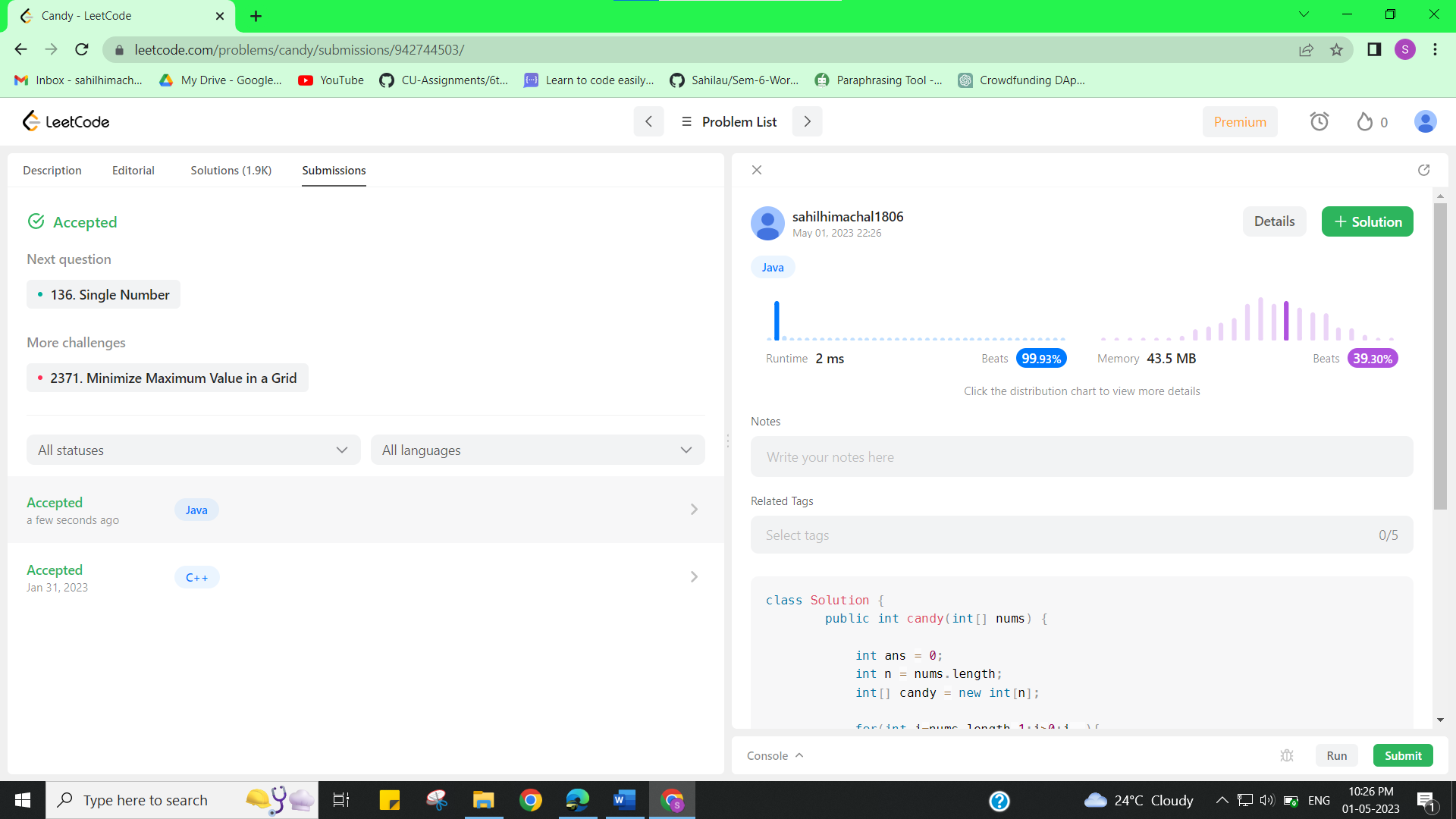
            return n+ans+candy[n-1];

        }

}

1. **Result/Output/Writing Summary:**

****

****

**Learning outcomes (What I have learnt):**

* Learned the concept of Greedy Approach.