

**UNIVERSITY INSTITUTE OF ENGINEERING**

**Department of Computer Science & Engineering**

# Subject Name: Competitive Coding 2

**Subject Code:** 20CSP-351

**Submitted to: Submitted by:**

Faculty name: Mr. Ankesh Gupta Name: Sahil Kaundal

UID: 21BCS8197

Section: 616

Group: A

**INDEX**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ex. No** | **List of Experiments** | **Conduct (MM: 12)** | **Viva**  **(MM: 10)** | **Record (MM: 8)** | **Total**  **(MM: 30)** | **Remarks/Signature** |
| 1. | Arrays, Stacks, Queues linked list |  |  |  |  |  |
| 2. | String Matching |  |  |  |  |  |
| 3. | Heap Model |  |  |  |  |  |
| 4. |  |  |  |  |  |  |
| 5. |  |  |  |  |  |  |
| 6. |  |  |  |  |  |  |
| 7. |  |  |  |  |  |  |
| 8. |  |  |  |  |  |  |
| 9. |  |  |  |  |  |  |
| 10. |  |  |  |  |  |  |

**Experiment 3.1**

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

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

**Semester:** 6th **Date of Performance:** 27/02/2023

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

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

Last Stone Weight

You are given an array of integers stones where stones[i] is the weight of the ith stone. We are playing a game with the stones. On each turn, we choose the **heaviest two stones** and smash them together. Suppose the heaviest two stones have weights x and y with x <= y. The result of this smash

<https://leetcode.com/problems/last-stone-weight/>

1. **Apparatus / Simulator Used:**

* Windows 7 or above
* Google Chrome

**3. Code:**

class Solution(object):

   def lastStoneWeight(self, stones):

      if len(stones) ==0:

         return 0

      if len(stones) == 1:

         return 1

      while len(stones)>1:

         stones.sort()

         s1,s2=stones[-1],stones[-2]

         if s1==s2:

            stones.pop(-1)

            stones.pop(-1)

         else:

            s1 = abs(s1-s2)

            stones.pop(-1)

            stones[-1] = s1

      if len(stones):

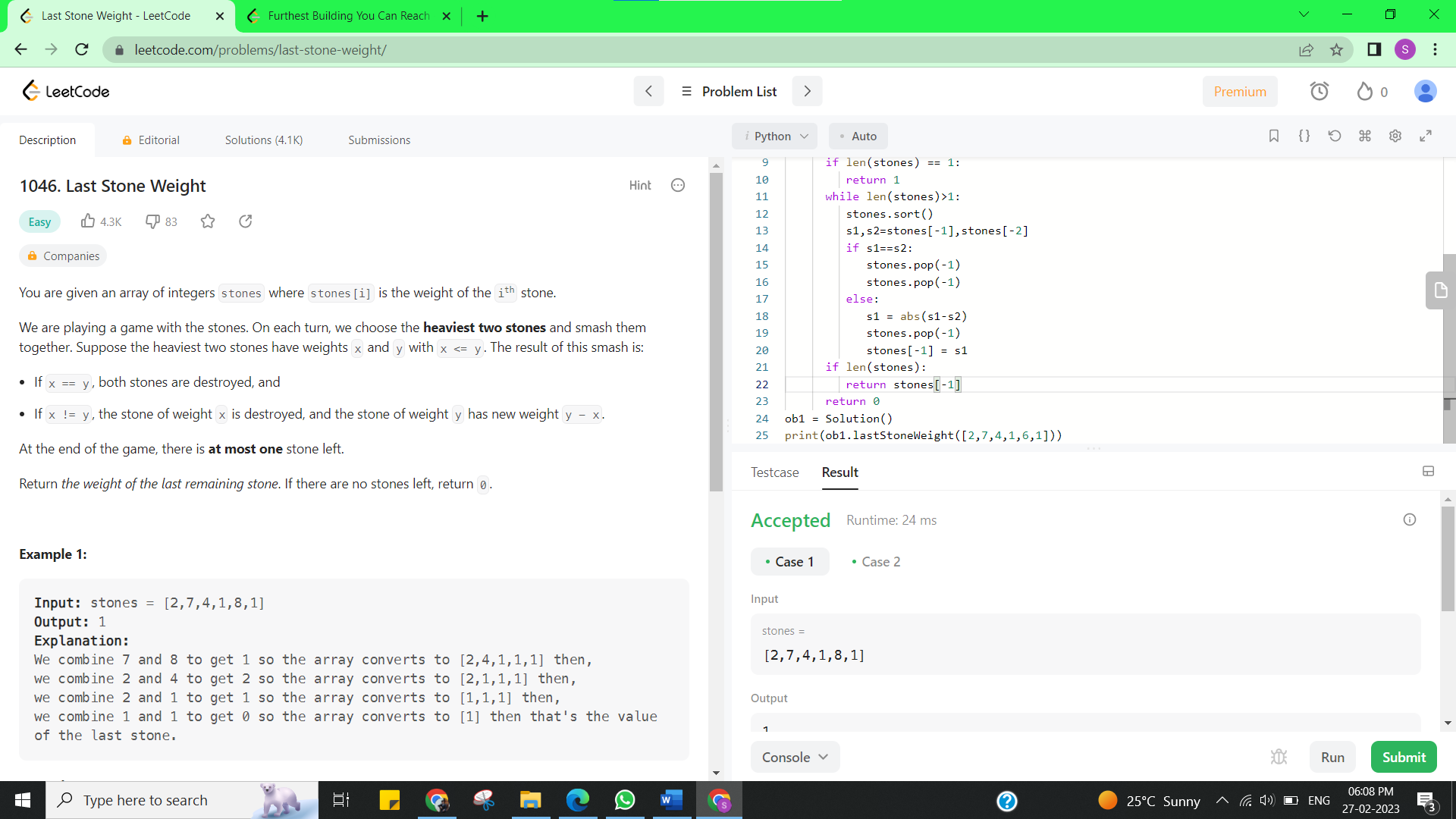
         return stones[-1]

      return 0

ob1 = Solution()

print(ob1.lastStoneWeight([2,7,4,1,6,1]))

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



**Experiment 3.2**

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

Furthest Building You Can Reach

You are given an integer array heights representing the heights of buildings, some bricks, and some ladders. You start your journey from building 0 and move to the next building by possibly using bricks or ladders.

<https://leetcode.com/problems/furthest-building-you-can-reach/>

1. **Apparatus / Simulator Used:**

* Windows 7 or above
* Google Chrome

1. **Code:**

class Solution {

public:

    int furthestBuilding(vector<int>& heights, int bricks, int ladders) {

        priority\_queue<int, vector<int>, greater<int>> min\_heap;

        for (int i = 0; i < size(heights) - 1; ++i) {

            int diff = heights[i + 1] - heights[i];

            if (diff > 0) {

                min\_heap.emplace(diff);

            }

            if (size(min\_heap) <= ladders) {  // ladders are reserved for largest diffs

                continue;

            }

            auto h = min\_heap.top(); min\_heap.pop();

            bricks -= h;  // use bricks if ladders are not enough

            if (bricks < 0) {  // not enough bricks

                return i;

            }

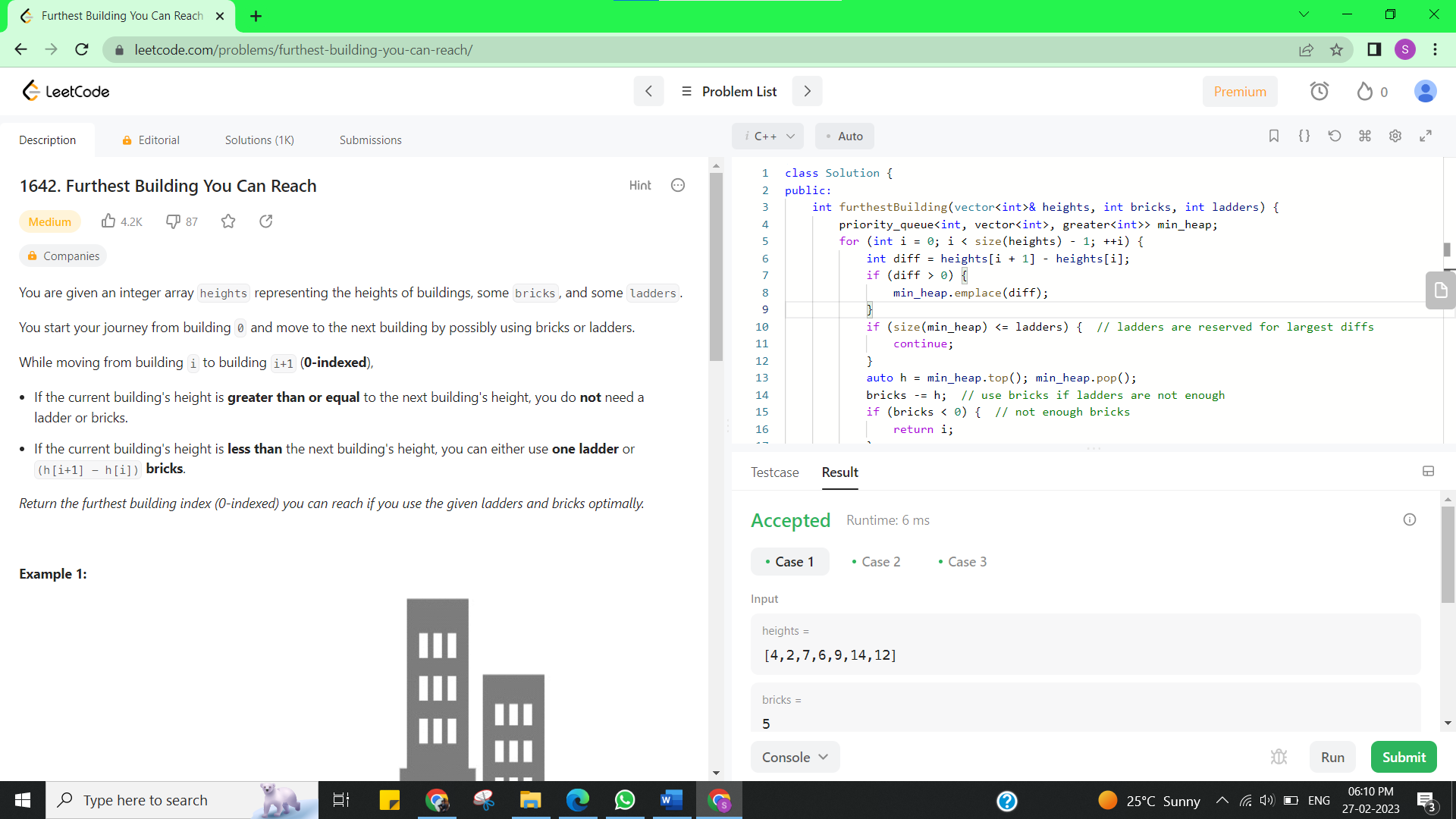
        }

        return size(heights) - 1;

    }

};

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

****

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

* + Learned the concept of Repeated string match.
  + Learnt about Furthest Building You Can Reach.

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |