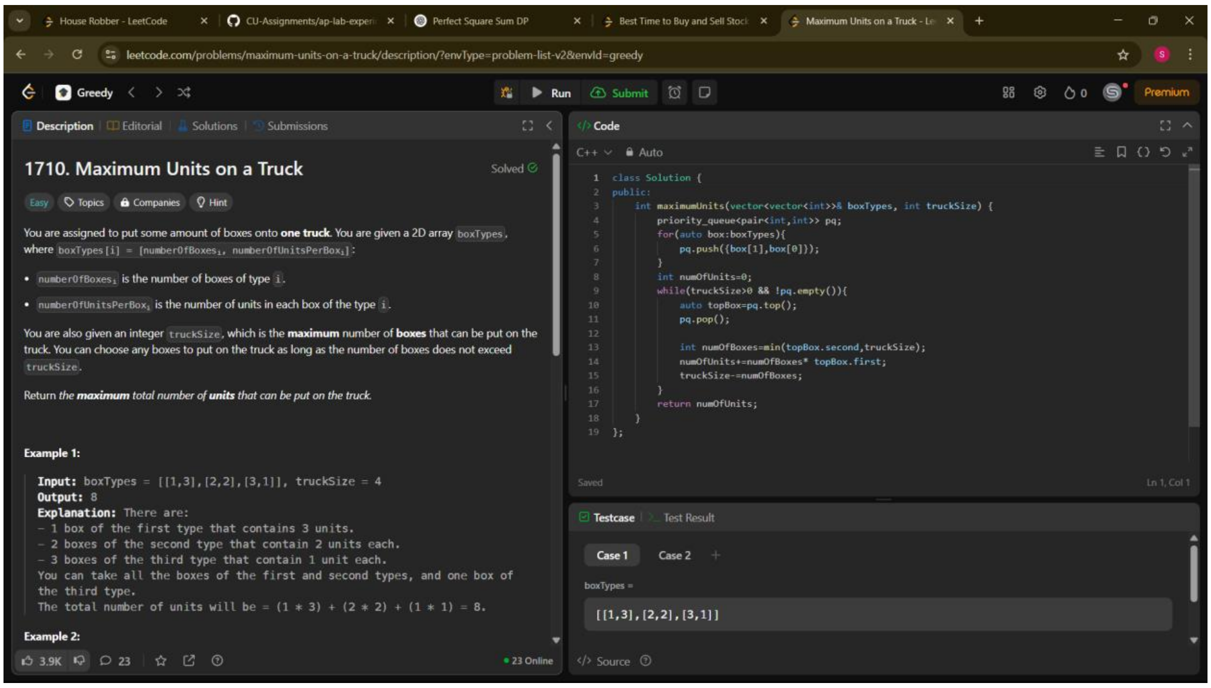
AP Experiment-8

Name: Shivangi Gupta UID: 22BCS15008 Section: 601-A

Q1. Maximum Units on a Truck https://leetcode.com/problems/maximum-units-on-a-truck/description/?envType=problem-list-v2&envId=greedy

CODE:

```
class Solution {
public:
    int maximumUnits(vector<vector<int>>& boxTypes, int truckSize) {
        priority_queue<pair<int,int>> pq;
        for(auto box:boxTypes){
            pq.push({box[1],box[0]});
        }
        int numOfUnits=0;
        while(truckSize>0 && !pq.empty()){
            auto topBox=pq.top();
            pq.pop();

            int numOfBoxes=min(topBox.second,truckSize);
            numOfUnits+=numOfBoxes* topBox.first;
            truckSize-=numOfBoxes;
        }
        return numOfUnits;
    }
};
```



leetcode.com/problems/maximum-units-on-a-truck/description/?envType=problem-list-v2&envId=greedy

1710. Maximum Units on a Truck

Solved

You are assigned to put some amount of boxes onto **one truck**. You are given a 2D array `boxTypes`, where `boxTypes[i] = [numberOfBoxes_i, numberOfUnitsPerBox_i]`:

- `numberOfBoxes_i` is the number of boxes of type `i`.
- `numberOfUnitsPerBox_i` is the number of units in each box of the type `i`.

You are also given an integer `truckSize`, which is the **maximum** number of **boxes** that can be put on the truck. You can choose any boxes to put on the truck as long as the number of boxes does not exceed `truckSize`.

Return *the **maximum** total number of **units** that can be put on the truck.*

Example 1:

```
Input: boxTypes = [[1,3],[2,2],[3,1]], truckSize = 4
Output: 8
Explanation: There are:
- 1 box of the first type that contains 3 units.
- 2 boxes of the second type that contain 2 units each.
- 3 boxes of the third type that contain 1 unit each.
You can take all the boxes of the first and second types, and one box of
the third type.
The total number of units will be = (1 * 3) + (2 * 2) + (1 * 1) = 8.
```

Example 2:

```cpp
class Solution {
public:
    int maximumUnits(vector<vector<int>>& boxTypes, int truckSize) {
        priority_queue<pair<int,int>> pq;
        for(auto box:boxTypes){
            pq.push({box[1],box[0]});
        }
        int numOfUnits=0;
        while(truckSize>0 && !pq.empty()){
            auto topBox=pq.top();
            pq.pop();

            int numOfBoxes=min(topBox.second,truckSize);
            numOfUnits+=numOfBoxes* topBox.first;
            truckSize-=numOfBoxes;
        }
        return numOfUnits;
    }
};
```

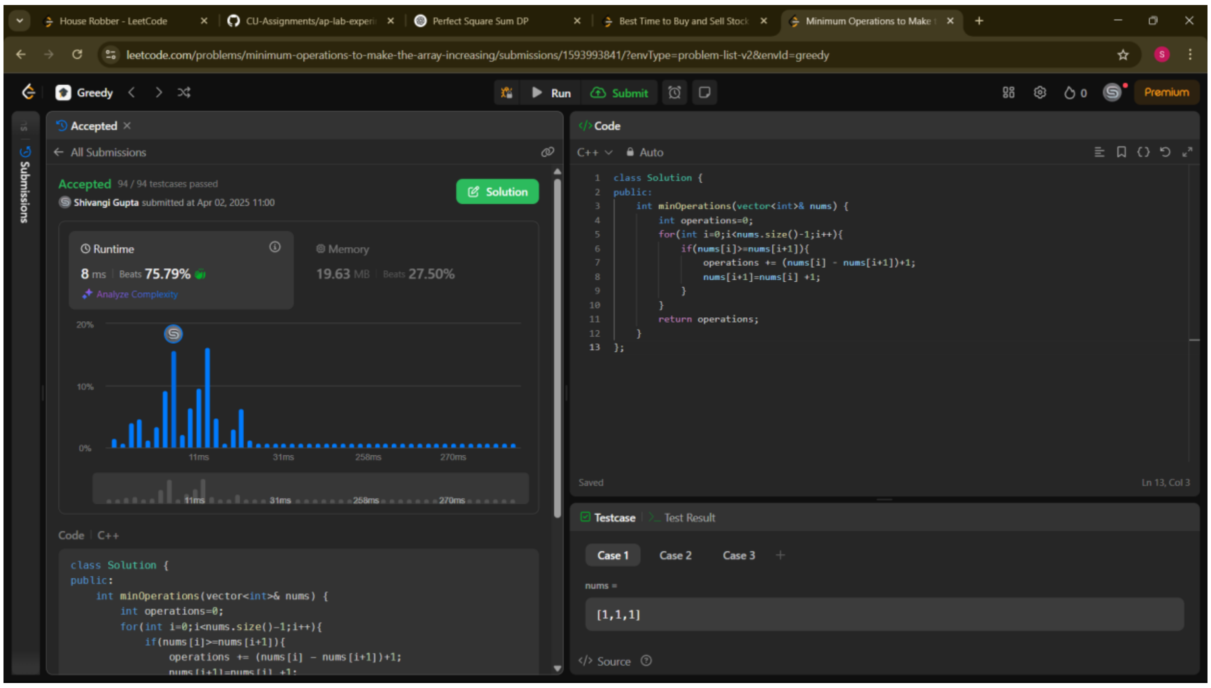
boxTypes =

```
[[1,3],[2,2],[3,1]]
```

Q2. https://leetcode.com/problems/minimum-operations-to-make-the-array-increasing/description/?envType=problem-list-v2&envId=greedy

CODE:

```
class Solution {

public:

   int minOperations(vector<int>& nums) {

      int operations=0;

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

         if(nums[i]>=nums[i+1]){

            operations += (nums[i] - nums[i+1])+1;

            nums[i+1]=nums[i] +1;

         }

      }

      return operations;

   }

};
```



leetcode.com/problems/minimum-operations-to-make-the-array-increasing/submissions/1593993841/?envType=problem-list-v2&envId=greedy

Accepted
All Submissions
Accepted 94 / 94 testcases passed
Shivangi Gupta submitted at Apr 02, 2025 11:00

Runtime
8 ms | Beats 75.79%

Memory
19.63 MB | Beats 27.50%

Code | C++

```cpp
class Solution {
public:
    int minOperations(vector<int>& nums) {
        int operations=0;
        for(int i=0;i<nums.size()-1;i++){
            if(nums[i]>=nums[i+1]){
                operations += (nums[i] - nums[i+1])+1;
                nums[i+1]=nums[i] +1;
            }
        }
        return operations;
    }
};
```

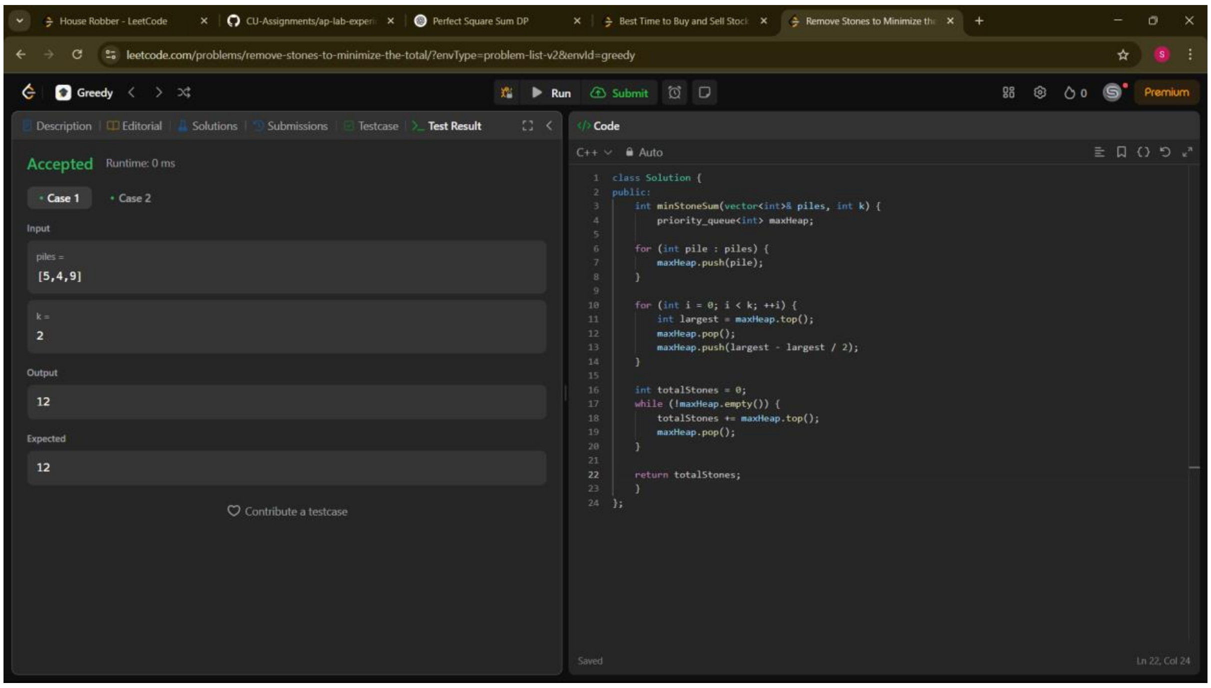
Saved
Ln 13, Col 3

Testcase | Test Result
Case 1 Case 2 Case 3
nums =
[1,1,1]

Q3. https://leetcode.com/problems/remove-stones-to-minimize-the-total/?envType=problem-list-v2&envId=greedy

CODE:

```cpp
class Solution {

public:

    int minStoneSum(vector<int>& piles, int k) {

        priority_queue<int> maxHeap;


        for (int pile : piles) {

            maxHeap.push(pile);

        }


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

            int largest = maxHeap.top();

            maxHeap.pop();

            maxHeap.push(largest - largest / 2);

        }


        int totalStones = 0;
    while (!maxHeap.empty()) {
        totalStones += maxHeap.top();
        maxHeap.pop();
    }

    return totalStones;
    }
};
```



leetcode.com/problems/remove-stones-to-minimize-the-total/?envType=problem-list-v2&envId=greedy

Accepted Runtime: 0 ms

Case 1 Case 2

Input

piles =
[5,4,9]

k =
2

Output

12

Expected

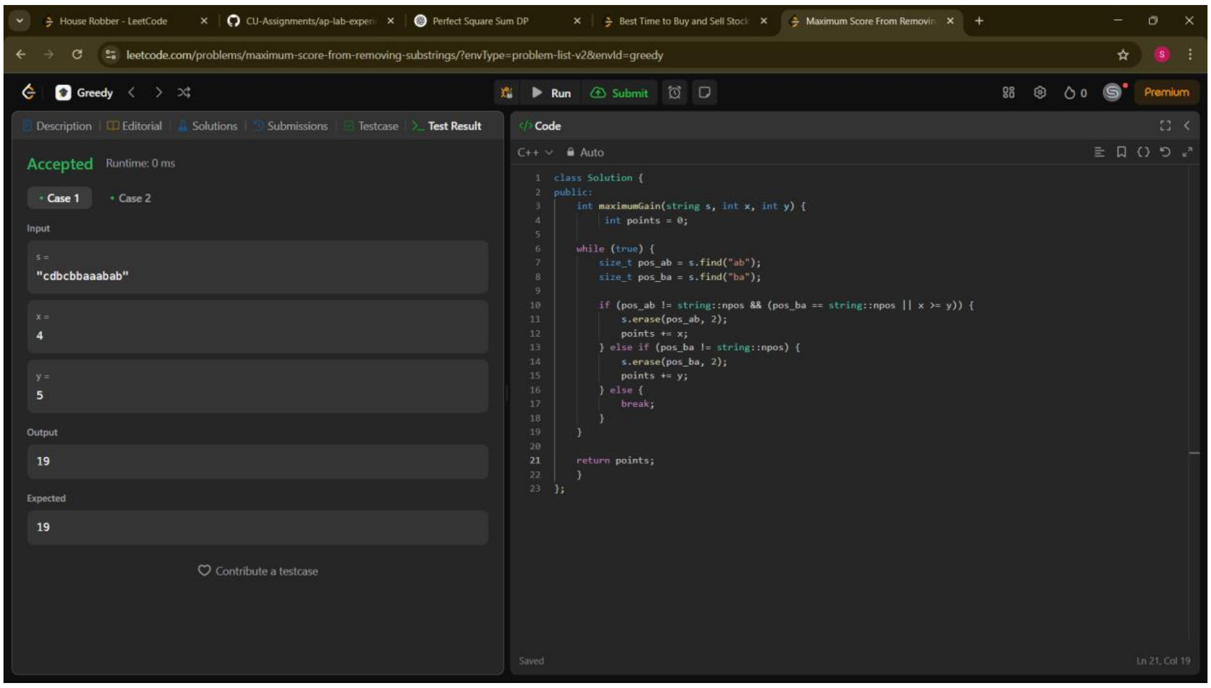
12

```
class Solution {
public:
    int minStoneSum(vector<int>& piles, int k) {
        priority_queue<int> maxHeap;

        for (int pile : piles) {
            maxHeap.push(pile);
        }

        for (int i = 0; i < k; ++i) {
            int largest = maxHeap.top();
            maxHeap.pop();
            maxHeap.push(largest - largest / 2);
        }

        int totalStones = 0;
        while (!maxHeap.empty()) {
            totalStones += maxHeap.top();
            maxHeap.pop();
        }

        return totalStones;
    }
};
```

Q4. https://leetcode.com/problems/maximum-score-from-removing-substrings/?envType=problem-list-v2&envId=greedy

CODE:

```
class Solution {
public:
    int maximumGain(string s, int x, int y) {
         int points = 0;

    while (true) {
        size_t pos_ab = s.find("ab");
        size_t pos_ba = s.find("ba");

        if (pos_ab != string::npos && (pos_ba == string::npos || x >= y)) {
            s.erase(pos_ab, 2);
            points += x;
        } else if (pos_ba != string::npos) {
            s.erase(pos_ba, 2);
            points += y;
        } else {
            break;
        }
    }

    return points;
    }
};
```


House Robber - LeetCode
CU-Assignments/ap-lab-experi
Perfect Square Sum DP
Best Time to Buy and Sell Stock
Maximum Score From Removin
leetcode.com/problems/maximum-score-from-removing-substrings/?envType=problem-list-v2&envId=greedy
Greedy
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
Testcase
Test Result
Code
C++
Auto
Contribute a testcase
Saved
Ln 21, Col 19

Accepted Runtime: 0 ms

Case 1 Case 2

Input

s =
"cdbcbbaaabab"

x =
4

y =
5

Output

19

Expected

19

```
class Solution {
public:
    int maximumGain(string s, int x, int y) {
        int points = 0;

    while (true) {
        size_t pos_ab = s.find("ab");
        size_t pos_ba = s.find("ba");

        if (pos_ab != string::npos && (pos_ba == string::npos || x >= y)) {
            s.erase(pos_ab, 2);
            points += x;
        } else if (pos_ba != string::npos) {
            s.erase(pos_ba, 2);
            points += y;
        } else {
            break;
        }
    }

    return points;
    }
};
```