NAME- Ashish kumar Singh

CLASS- 22BCS-IOT-614/B

UID – 22BCS16892

SUBJECT- AP LAB

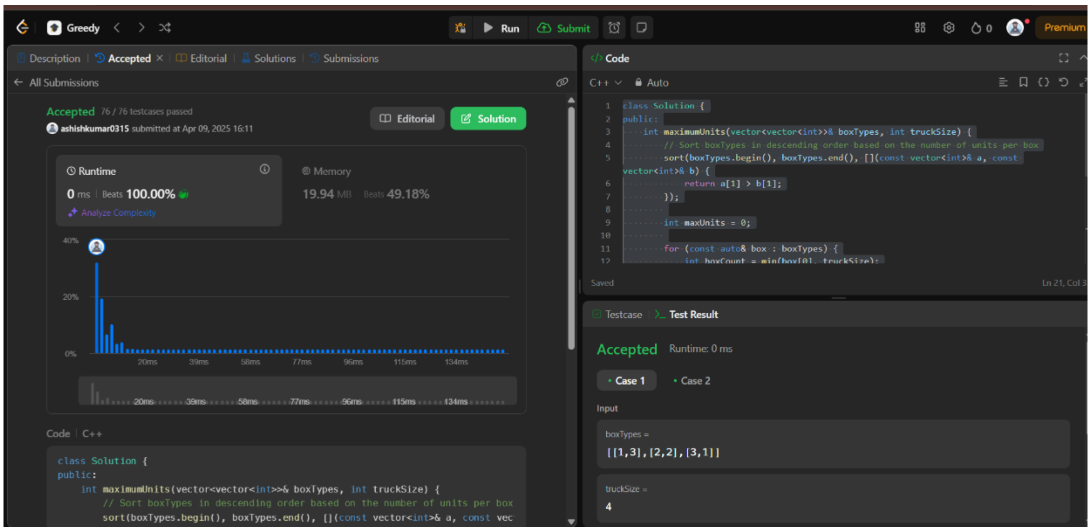
QUESTION -1:

1710. Maximum Units on a Truck

CODE:

```
class Solution {
public:
    int maximumUnits(vector<vector<int>>& boxTypes, int truckSize)
        sort(boxTypes.begin(), boxTypes.end(), [](const vector<int>& a, const vector<int>& b) {
            return a[1] > b[1];
        });
        int maxUnits = 0;
        for (const auto& box : boxTypes) {
            int boxCount = min(box[0], truckSize);
            maxUnits += boxCount * box[1];
            truckSize -= boxCount;
            if (truckSize == 0) break;
        }
        return maxUnits;
    }
};
```

OUTPUT:



Accepted 76 / 76 testcases passed
ashishkumar0315 submitted at Apr 09, 2025 16:11

Runtime
0 ms | Beats 100.00%

Memory
19.94 MB | Beats 49.18%

Code | C++

```
class Solution {
public:
    int maximumUnits(vector<vector<int>>& boxTypes, int truckSize) {
        // Sort boxTypes in descending order based on the number of units per box
        sort(boxTypes.begin(), boxTypes.end(), [](const vector<int>& a, const vec
```

```
class Solution {
public:
    int maximumUnits(vector<vector<int>>& boxTypes, int truckSize) {
        // Sort boxTypes in descending order based on the number of units per box
        sort(boxTypes.begin(), boxTypes.end(), [](const vector<int>& a, const 
vector<int>& b) {
            return a[1] > b[1];
        });
        
        int maxUnits = 0;
        
        for (const auto& box : boxTypes) {
            int boxCount = min(box[0], truckSize);
```

Accepted Runtime: 0 ms

Input

boxTypes =
[[1,3],[2,2],[3,1]]

truckSize =
4

QUESTION -2:

1827. Minimum Operations to Make the Array Increasing

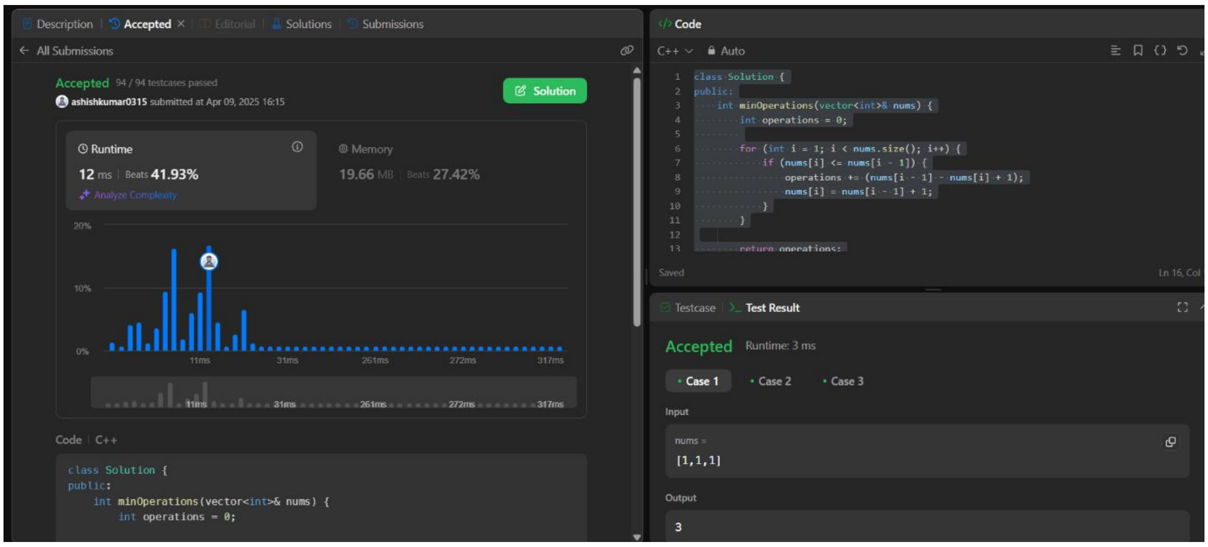
CODE:

```
class Solution {
public:
    int minOperations(vector<int>& nums) {
        int operations = 0;

        for (int i = 1; i < nums.size(); i++) {
            if (nums[i] <= nums[i - 1]) {
                operations += (nums[i - 1] - nums[i] + 1);
                nums[i] = nums[i - 1] + 1;
            }
        }

        return operations;
    }
};
```

OUTPUT:



Accepted 94 / 94 testcases passed

ashishkumar0315 submitted at Apr 09, 2025 16:15

Runtime 12 ms | Beats 41.93%

Memory 19.66 MB | Beats 27.42%

Code | C++

```
class Solution {
public:
    int minOperations(vector<int>& nums) {
        int operations = 0;
```

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

        return operations;
```

Accepted Runtime: 3 ms

Input

```
nums =
[1,1,1]
```

Output

```
3
```

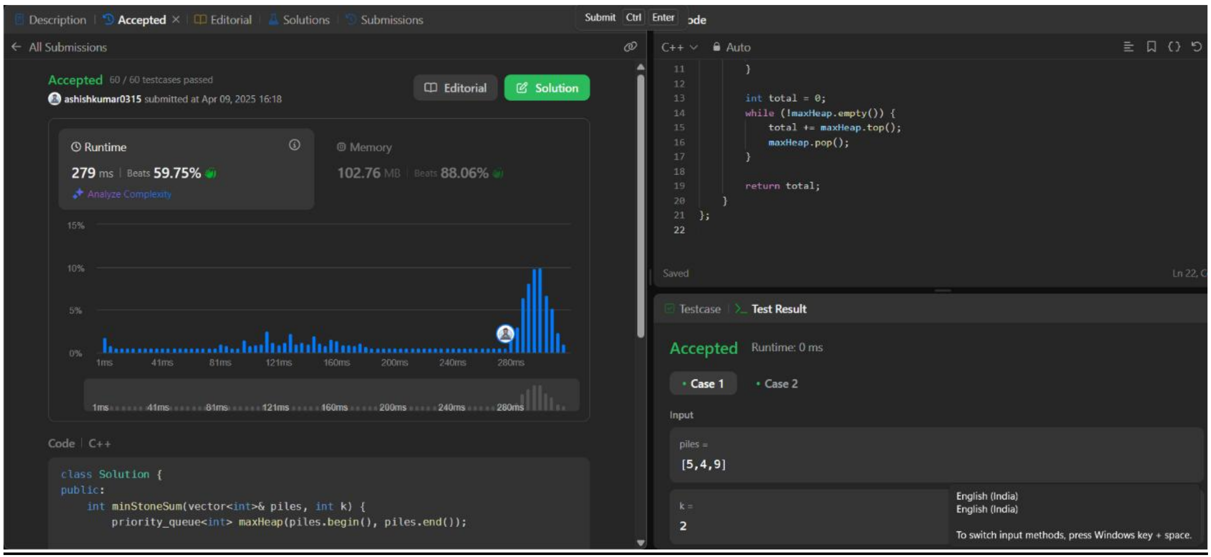
QUESTION -3:

1962. Remove Stones to Minimize the Total

CODE:

```
class Solution {
public:
  int minStoneSum(vector<int>& piles, int k) {
    priority_queue<int> maxHeap(piles.begin(), piles.end());
    while (k--) {
      int top = maxHeap.top();
      maxHeap.pop();
      top -= top / 2;
      maxHeap.push(top);
    }
    int total = 0;
    while (!maxHeap.empty()) {
      total += maxHeap.top();
      maxHeap.pop();
    }
    return total;
  }};
```

OUTPUT:



Accepted 60 / 60 testcases passed
ashishkumar0315 submitted at Apr 09, 2025 16:18

Runtime
279 ms | Beats 59.75%

Memory
102.76 MB | Beats 88.06%

Code | C++

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

```
        }

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

        return total;
    }
};
```

Saved

Ln 22, C

Accepted Runtime: 0 ms

Input

piles =
[5,4,9]

k =
2

English (India)
English (India)

To switch input methods, press Windows key + space.

QUESTION -4:

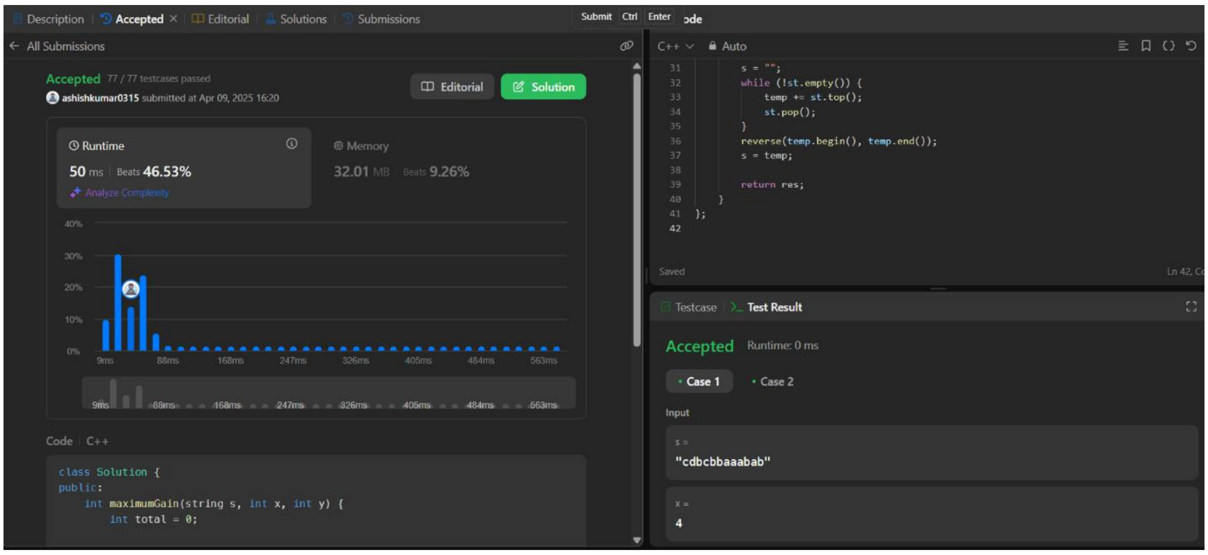
[1717. Maximum Score From Removing Substrings]

CODE:

```
class Solution {
public:
    int maximumGain(string s, int x, int y) {
        int total = 0;
        if (x > y) {
            total += removePair(s, 'a', 'b', x);
            total += removePair(s, 'b', 'a', y);
        } else {
            total += removePair(s, 'b', 'a', y);
            total += removePair(s, 'a', 'b', x);
        }
        return total;
    }
    int removePair(string& s, char first, char second, int score) {
        stack<char> st;
        int res = 0;
        string temp = "";
        for (char c : s) {
```

```
if (!st.empty() && st.top() == first && c == second) {
            st.pop();
            res += score;
        } else {
            st.push(c);
        } }
    s = "";
    while (!st.empty()) {
        temp += st.top();
        st.pop();
    }
    reverse(temp.begin(), temp.end());
    s = temp;
    return res;
}};
```

OUTPUT:



Accepted 77 / 77 testcases passed
ashishkumar0315 submitted at Apr 09, 2025 16:20

50 ms | Beats 46.53%

32.01 MB | Beats 9.26%

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

```
        s = "";
        while (!st.empty()) {
            temp += st.top();
            st.pop();
        }
        reverse(temp.begin(), temp.end());
        s = temp;

        return res;
    }
};
```

Accepted Runtime: 0 ms

s =
"cdbcbbaaabab"

x =
4

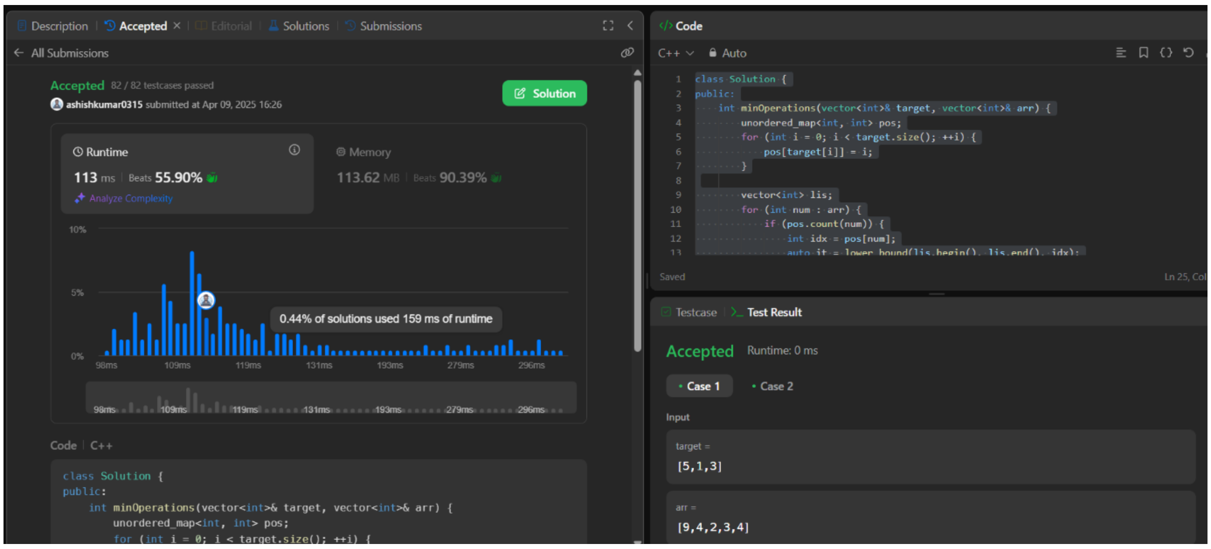
QUESTION -5:

1713. Minimum Operations to Make a Subsequence

CODE:

```
class Solution {
public:
  int minOperations(vector<int>& target, vector<int>& arr) {
    unordered_map<int, int> pos;
    for (int i = 0; i < target.size(); ++i) {
      pos[target[i]] = i;
    }
    vector<int> lis;
    for (int num : arr) {
      if (pos.count(num)) {
        int idx = pos[num];
        auto it = lower_bound(lis.begin(), lis.end(), idx);
        if (it == lis.end()) {
          lis.push_back(idx);
        } else {
          *it = idx;
        }    }   }
    return target.size() - lis.size();
  }
};
```

OUTPUT:



Accepted 82 / 82 testcases passed
ashishkumar0315 submitted at Apr 09, 2025 16:26

Runtime
113 ms | Beats 55.90%

Memory
113.62 MB | Beats 90.39%

0.44% of solutions used 159 ms of runtime

Code | C++

```
class Solution {
public:
    int minOperations(vector<int>& target, vector<int>& arr) {
        unordered_map<int, int> pos;
        for (int i = 0; i < target.size(); ++i) {
```

```
class Solution {
public:
    int minOperations(vector<int>& target, vector<int>& arr) {
        unordered_map<int, int> pos;
        for (int i = 0; i < target.size(); ++i) {
            pos[target[i]] = i;
        }

        vector<int> lis;
        for (int num : arr) {
            if (pos.count(num)) {
                int idx = pos[num];
                auto it = lower_bound(lis.begin(), lis.end(), idx);
```

Saved
Ln 25, Col

Accepted Runtime: 0 ms

Input

target =
[5,1,3]

arr =
[9,4,2,3,4]

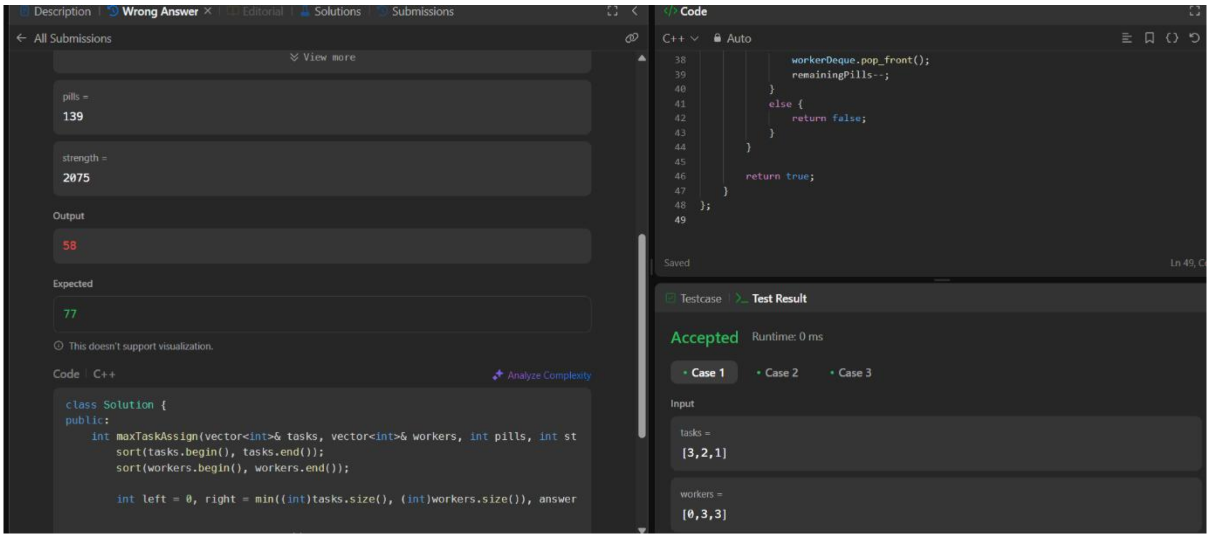
QUESTION -6:

2071. Maximum Number of Tasks You Can Assign

CODE:

```
class Solution {
public:
    int maxTaskAssign(vector<int>& tasks, vector<int>& workers, int pills, int strength) {
        sort(tasks.begin(), tasks.end());
        sort(workers.begin(), workers.end());
        int left = 0, right = min((int)tasks.size(), (int)workers.size());
        int result = 0;
        while (left <= right) {
            int mid = (left + right) / 2;
            if (canAssign(mid, tasks, workers, pills, strength)) {
                result = mid;
                left = mid + 1;
            } else {
                right = mid - 1;
            }    }
      return result;
    }
private:
    bool canAssign(int k, vector<int>& tasks, vector<int>& workers, int pills, int strength) {
        deque<int> dq(workers.end() - k, workers.end());
        multiset<int> taskSet(tasks.begin(), tasks.begin() + k);
        int remainingPills = pills;
        while (!taskSet.empty()) {
            int task = *taskSet.rbegin();
            taskSet.erase(--taskSet.end());
            if (!dq.empty() && dq.back() >= task) {
                dq.pop_back();
            } else if (!dq.empty() && dq.front() + strength >= task && remainingPills > 0) {
                dq.pop_front();
                remainingPills--;
          } else {
                return false;
            }    }
        return true;
    }};
```

OUTPUT:



Description | Wrong Answer | Editorial | Solutions | Submissions

All Submissions

View more

pills =
139

strength =
2075

Output
58

Expected
77

This doesn't support visualization.

Code | C++ Analyze Complexity

```
class Solution {
public:
    int maxTaskAssign(vector<int>& tasks, vector<int>& workers, int pills, int st
        sort(tasks.begin(), tasks.end());
        sort(workers.begin(), workers.end());

        int left = 0, right = min((int)tasks.size(), (int)workers.size()), answer
```

Code

C++ Auto

```
                    workerDeque.pop_front();
                    remainingPills--;
                }
                else {
                    return false;
                }
            }

        return true;
    }
};
```

Saved Ln 49,

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

tasks =
[3,2,1]

workers =
[0,3,3]