



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 8

Student Name: Ashish Kumar

Branch: CSE

Semester: 6

Subject Name: AP Lab

UID:22bcs11958

Section/Group:614(B)

Date of Performance:04/04/25

Subject Code: 22CSP-351

Q1:-Max Units on a Truck

```
class Solution {
public:
    int maximumUnits(vector<vector<int>>& boxTypes, int truckSize) {

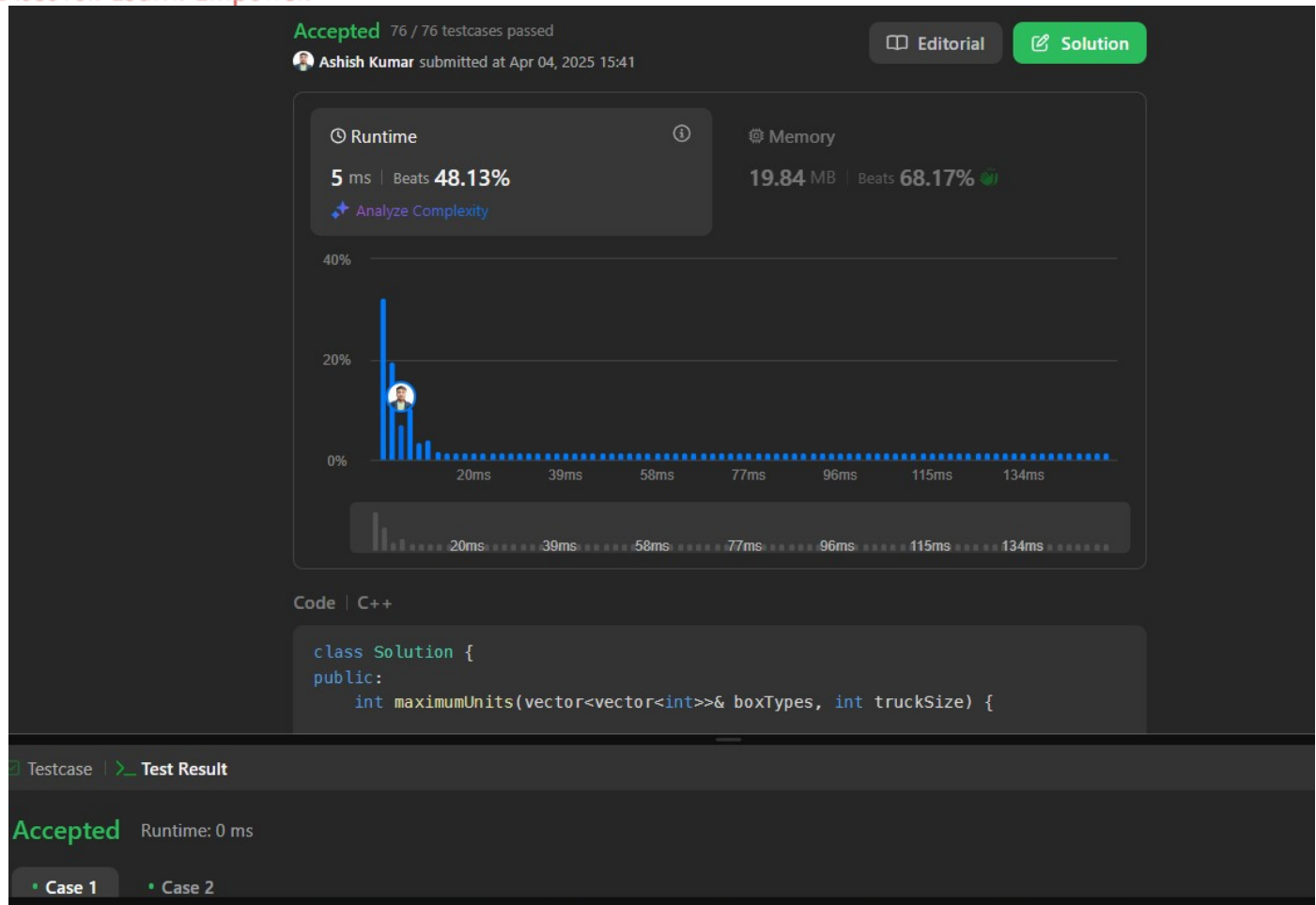
        sort(boxTypes.begin(), boxTypes.end(), [](const vector<int>& a, const vector<int>& b) {
            return a[1] > b[1];
        });
        int ans=0;
        for (const auto& box : boxTypes) {
            int boxesToTake = min(truckSize, box[0]);
            ans += boxesToTake * box[1];
            truckSize -= boxesToTake;

            if (truckSize == 0)
                break;
        }
        return ans;
    }
};
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.



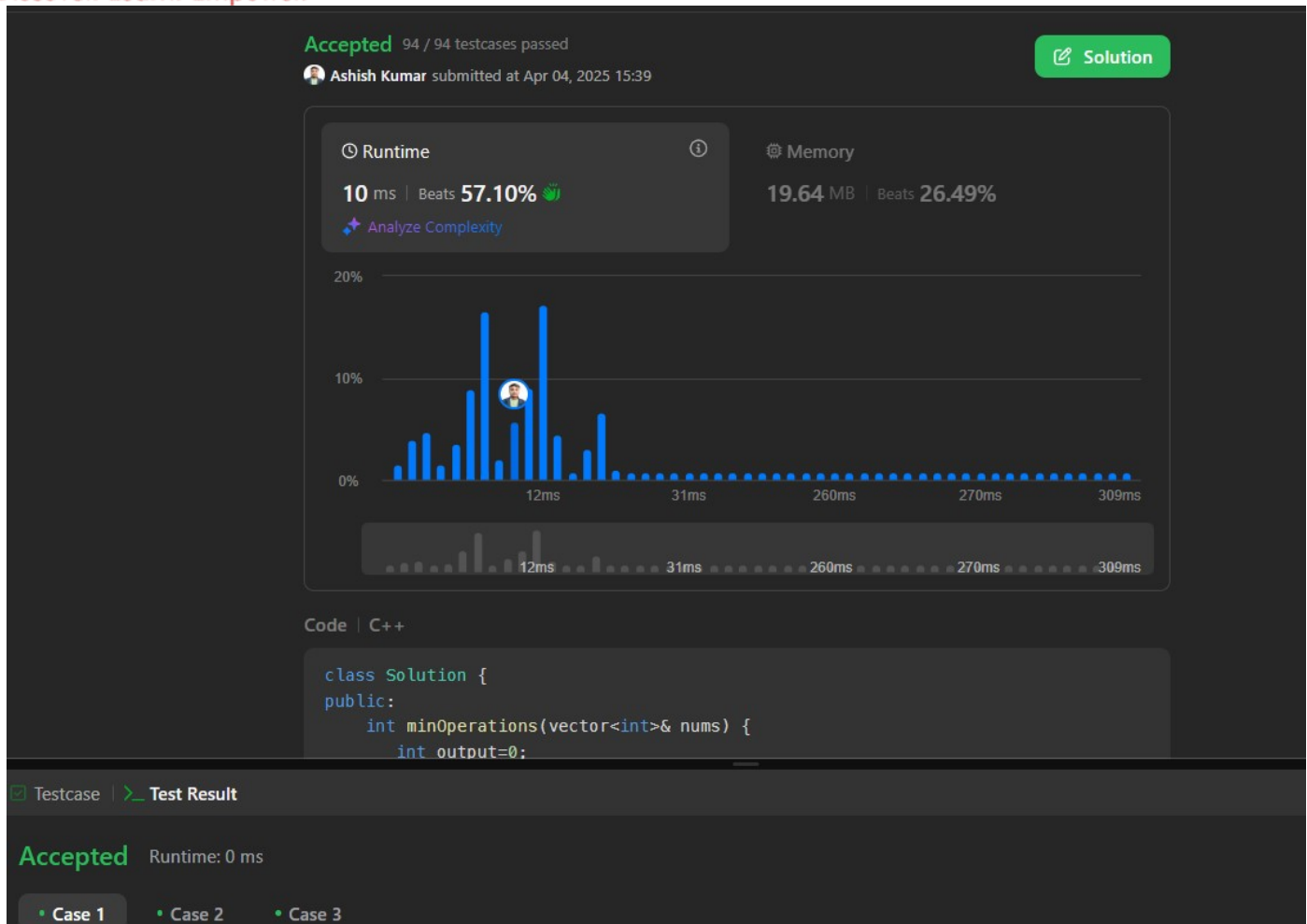
Q2:-Min Operations to Make Array Increasing

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



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.



Q3:-Max Score from Removing Substrings

```
class Solution {
public:
    int removeSubStr(string &s, string &matchStr) {
        stack<char> st;
        int removedCount = 0;

        for (char &ch : s) {
            if (ch == matchStr[1] && !st.empty() && st.top() == matchStr[0]) {
                st.pop();
                removedCount += 2;
            } else {
                st.push(ch);
            }
        }
        string temp;
        while (!st.empty()) {
            temp.push_back(st.top());
            st.pop();
        }
        reverse(temp.begin(), temp.end());
```

```
s = temp;

return removedCount;
}

int maximumGain(string s, int x, int y) {
    int n = s.length();
    int score = 0;

    string maxstr = (x >= y) ? "ab" : "ba";
    string minstr = (x < y) ? "ab" : "ba";

    int charRemove = removeSubStr(s, maxstr);
    score += (charRemove / 2) * max(x, y);

    charRemove = removeSubStr(s, minstr);
    score += (charRemove / 2) * min(x, y);

    return score;
};
```

← All Submissions

Accepted 77 / 77 testcases passed

Ashish Kumar submitted at Feb 21, 2025 17:08

Editorial

Solution

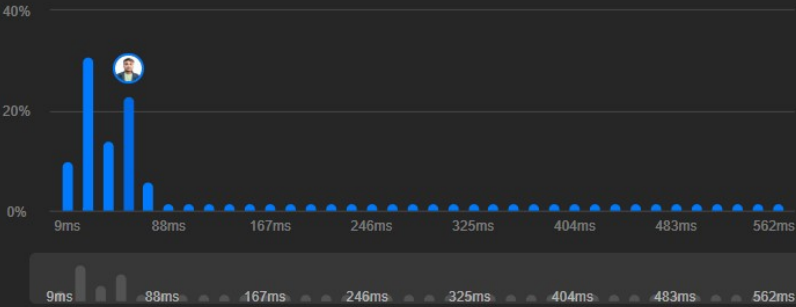
Runtime

69 ms | Beats 20.62%

Analyze Complexity

Memory

31.92 MB | Beats 9.90%



Code | C++

```
class Solution {
public:
    int removeSubStr(string &s, string &matchStr) {
        stack<char> st;
```

Testcase

Test Result

Accepted

Runtime: 0 ms

Case 1

Case 2



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Q4:-Remove Stones to Maximize Total

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

        while (k-->0) {
            int largest = maxHeap.top();
            maxHeap.pop();
            largest -= largest / 2;
            maxHeap.push(largest);
        }

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

        return total;
    }
};
```

</> Code | Accepted ×

← All Submissions

Accepted 60 / 60 testcases passed

Ashish Kumar submitted at Apr 11, 2025 22:55

Editorial Solution

Runtime

292 ms | Beats 48.36%

Analyze Complexity

Memory

102.72 MB | Beats 87.76%

Code | C++

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

Testcase > Test Result



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.