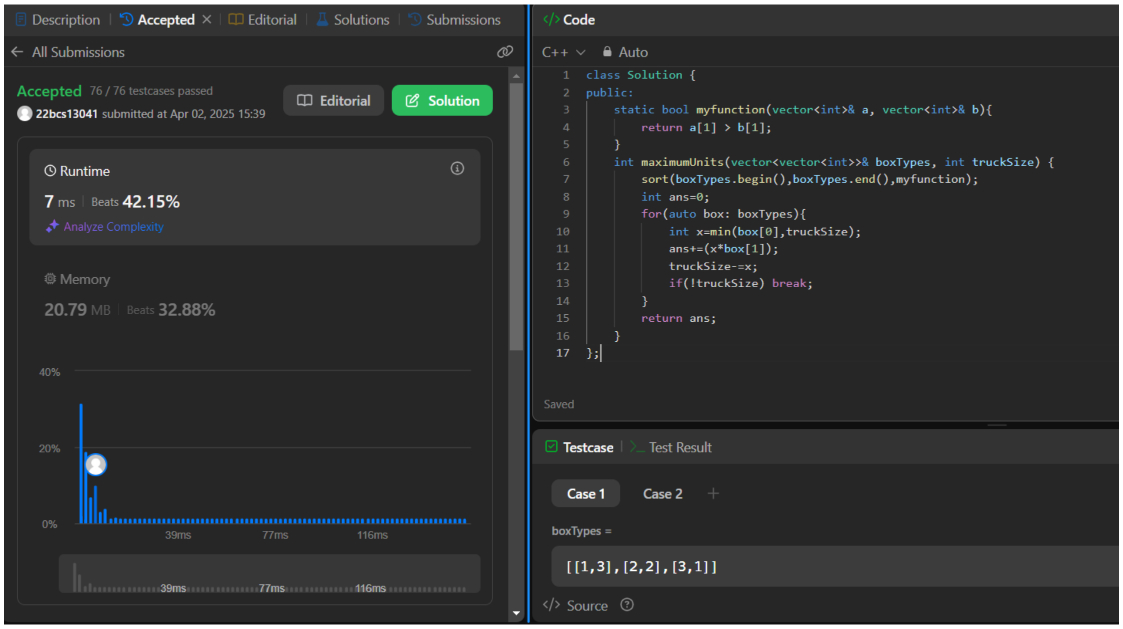
Name: Akash Pandey **UID:** 22BCS11135
Sec: FL_IOT-601/ A **Sub:** AP Lab -II

Max Units on a Truck

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



Accepted 76 / 76 testcases passed
22bcs13041 submitted at Apr 02, 2025 15:39

Runtime
7 ms | Beats 42.15%

Memory
20.79 MB | Beats 32.88%

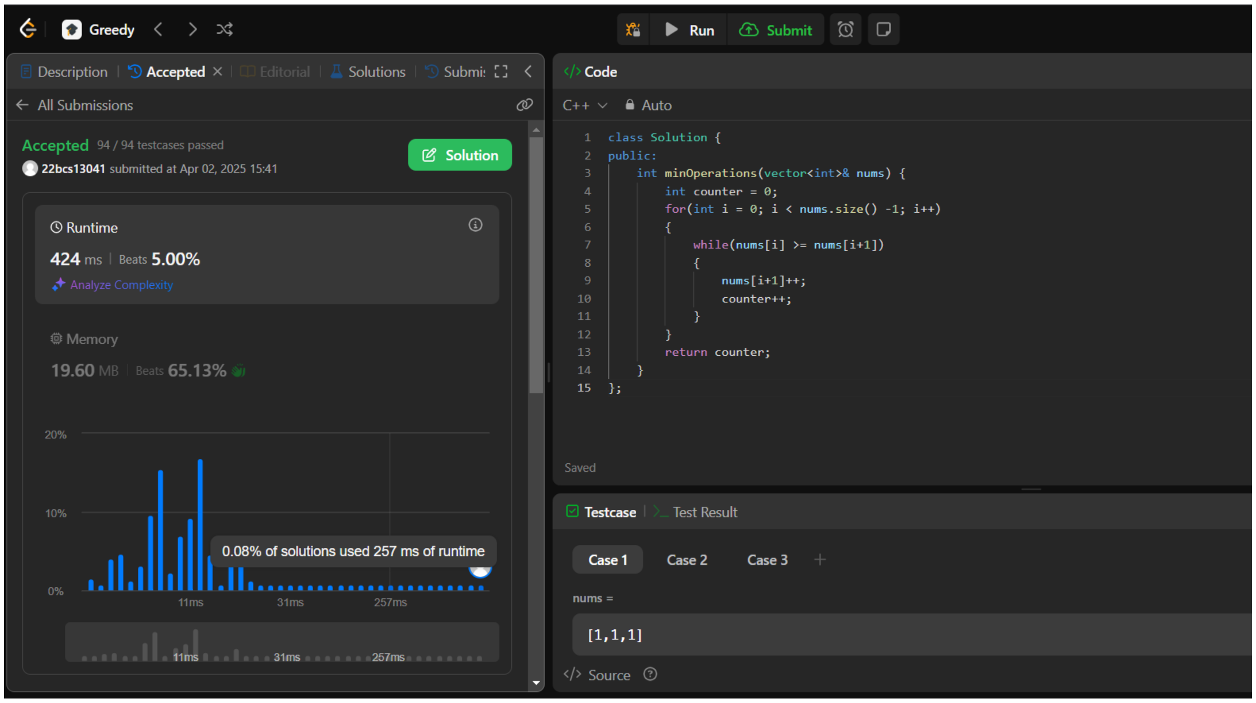
```
class Solution {
public:
    static bool myfunction(vector<int>& a, vector<int>& b){
        return a[1] > b[1];
    }
    int maximumUnits(vector<vector<int>>& boxTypes, int truckSize) {
        sort(boxTypes.begin(),boxTypes.end(),myfunction);
        int ans=0;
        for(auto box: boxTypes){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

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

Min Operations to Make Array Increasing

```
class Solution {
```

```
public:
    int minOperations(vector<int>& nums) {
        int counter = 0;
        for(int i = 0; i < nums.size() -1; i++)
        {
            while(nums[i] >= nums[i+1])
            {
                nums[i+1]++;
                counter++;
            }
        }
        return counter;
    }
};
```



Accepted 94 / 94 testcases passed

22bcs13041 submitted at Apr 02, 2025 15:41

Runtime

424 ms | Beats 5.00%

Memory

19.60 MB | Beats 65.13%

0.08% of solutions used 257 ms of runtime

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

Saved

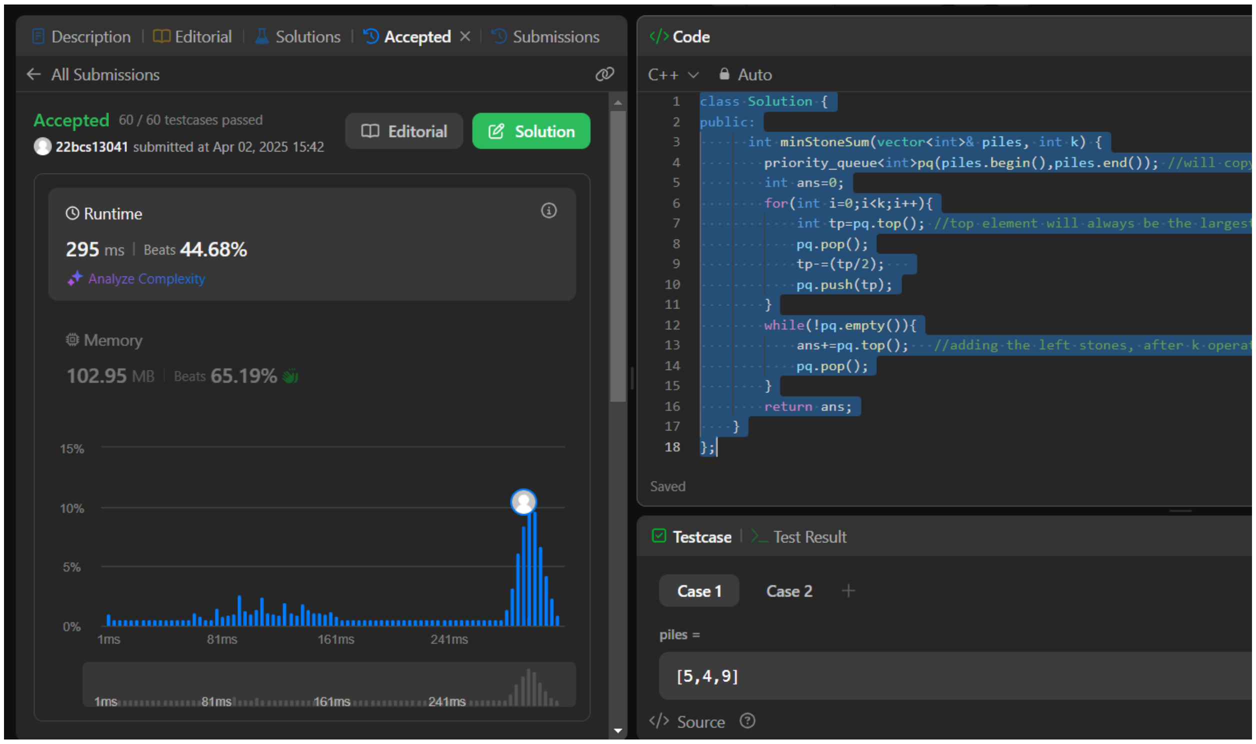
nums =

```
[1,1,1]
```

<u>Remove Stones to Maximize Total</u>

```
class Solution {
```

```
public:
    int minStoneSum(vector<int>& piles, int k) {
      priority_queue<int>pq(piles.begin(),piles.end()); //will copy the vector to the priority queue
      int ans=0;
      for(int i=0;i<k;i++){
        int tp=pq.top(); //top element will always be the largest element
        pq.pop();
        tp-=(tp/2);
        pq.push(tp);
      }
      while(!pq.empty()){
        ans+=pq.top();   //adding the left stones, after k operations
        pq.pop();
      }
      return ans;
    }
};
```



Accepted 60 / 60 testcases passed

22bcs13041 submitted at Apr 02, 2025 15:42

Runtime

295 ms | Beats 44.68%

Analyze Complexity

Memory

102.95 MB | Beats 65.19%

```
class Solution {
public:
    int minStoneSum(vector<int>& piles, int k) {
        priority_queue<int>pq(piles.begin(),piles.end()); //will copy
        int ans=0;
        for(int i=0;i<k;i++){
            int tp=pq.top(); //top element will always be the larges
            pq.pop();
            tp-=(tp/2);
            pq.push(tp);
        }
        while(!pq.empty()){
            ans+=pq.top();   //adding the left stones, after k operat
            pq.pop();
        }
        return ans;
    }
};
```

Testcase

Case 1

piles =

```
[5,4,9]
```

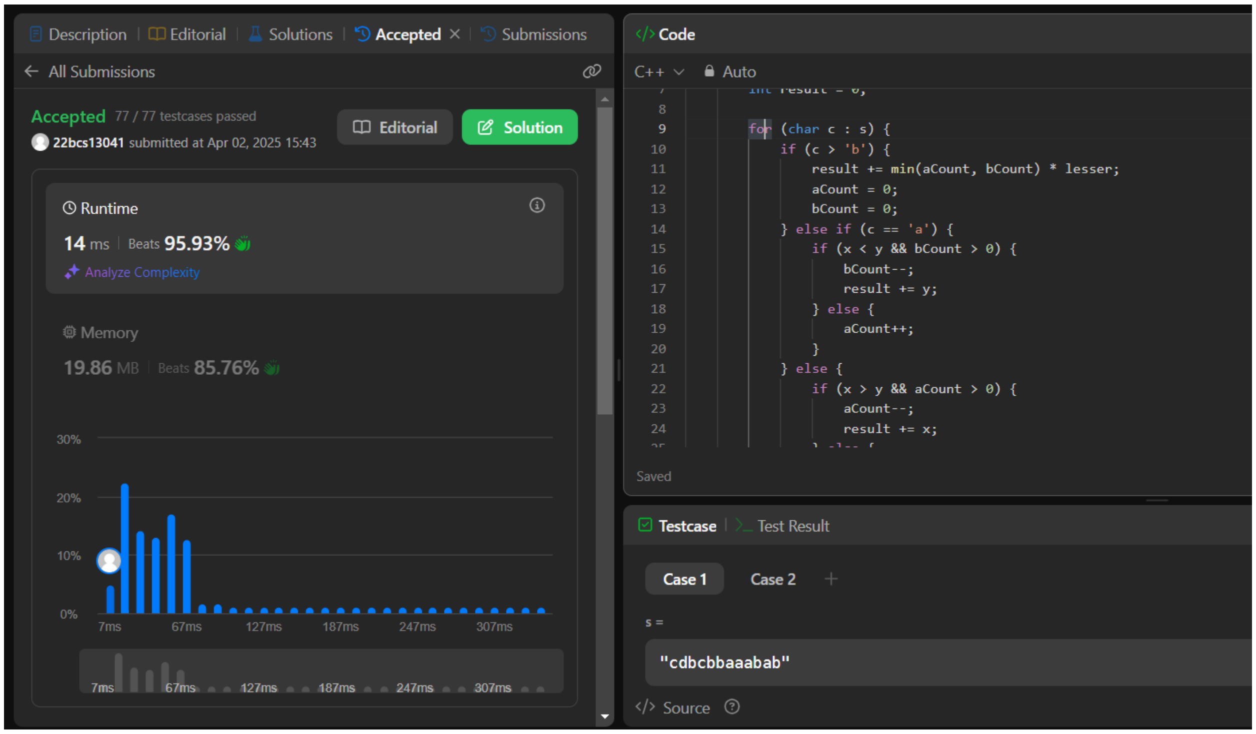
Max Score from Removing Substrings

```
class Solution {
```

```
public:
    int maximumGain(string s, int x, int y) {
        int aCount = 0;
        int bCount = 0;
        int lesser = min(x, y);
        int result = 0;

        for (char c : s) {
            if (c > 'b') {
                result += min(aCount, bCount) * lesser;
                aCount = 0;
                bCount = 0;
            } else if (c == 'a') {
                if (x < y && bCount > 0) {
                    bCount--;
                    result += y;
                } else {
                    aCount++;
                }
            } else {
                if (x > y && aCount > 0) {
                    aCount--;
                    result += x;
                } else {
                    bCount++;
                }
            }
        }

        result += min(aCount, bCount) * lesser;
        return result;
    }
};
```



Accepted 77 / 77 testcases passed
22bcs13041 submitted at Apr 02, 2025 15:43

Runtime
14 ms | Beats 95.93%

Memory
19.86 MB | Beats 85.76%

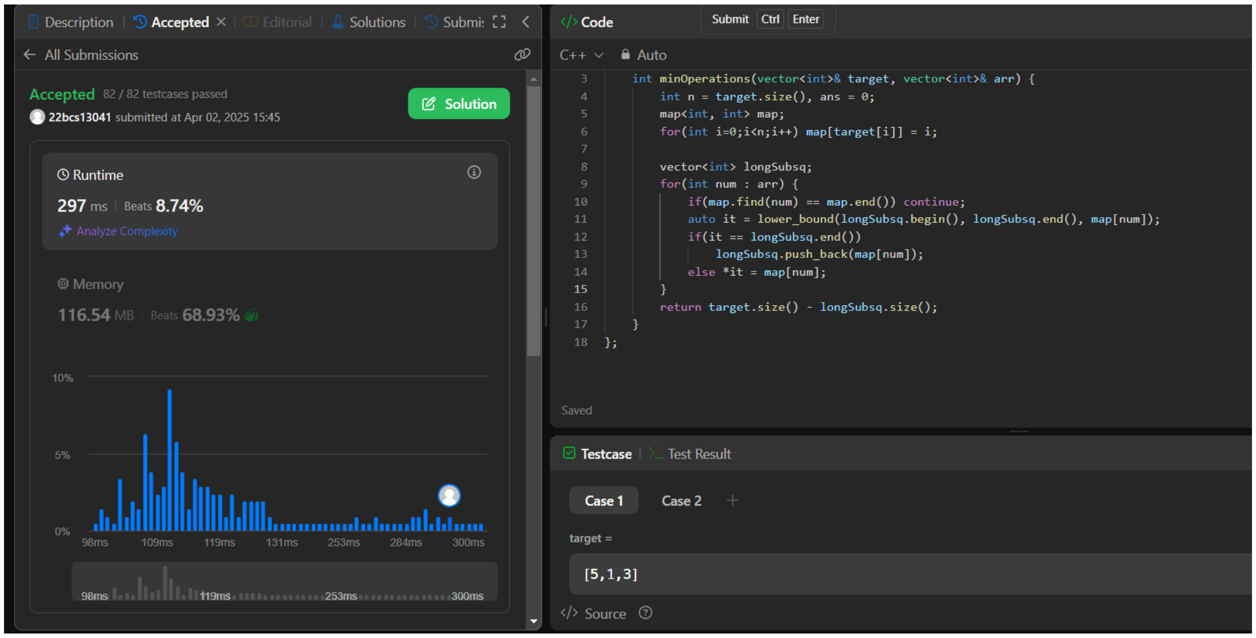
```
        for (char c : s) {
            if (c > 'b') {
                result += min(aCount, bCount) * lesser;
                aCount = 0;
                bCount = 0;
            } else if (c == 'a') {
                if (x < y && bCount > 0) {
                    bCount--;
                    result += y;
                } else {
                    aCount++;
                }
            } else {
                if (x > y && aCount > 0) {
                    aCount--;
                    result += x;
```

s =
"cdbcbbaaabab"

Min Operations to Make a Subsequence

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

        vector<int> longSubsq;
        for(int num : arr) {
            if(map.find(num) == map.end()) continue;
            auto it = lower_bound(longSubsq.begin(), longSubsq.end(), map[num]);
            if(it == longSubsq.end())
                longSubsq.push_back(map[num]);
            else *it = map[num];
        }
        return target.size() - longSubsq.size();
    }
};
```



Accepted 82 / 82 testcases passed

22bcs13041 submitted at Apr 02, 2025 15:45

297 ms | Beats 8.74%

116.54 MB | Beats 68.93%

```
    int minOperations(vector<int>& target, vector<int>& arr) {
        int n = target.size(), ans = 0;
        map<int, int> map;
        for(int i=0;i<n;i++) map[target[i]] = i;

        vector<int> longSubsq;
        for(int num : arr) {
            if(map.find(num) == map.end()) continue;
            auto it = lower_bound(longSubsq.begin(), longSubsq.end(), map[num]);
            if(it == longSubsq.end())
                longSubsq.push_back(map[num]);
            else *it = map[num];
        }
        return target.size() - longSubsq.size();
    }
};
```

target =

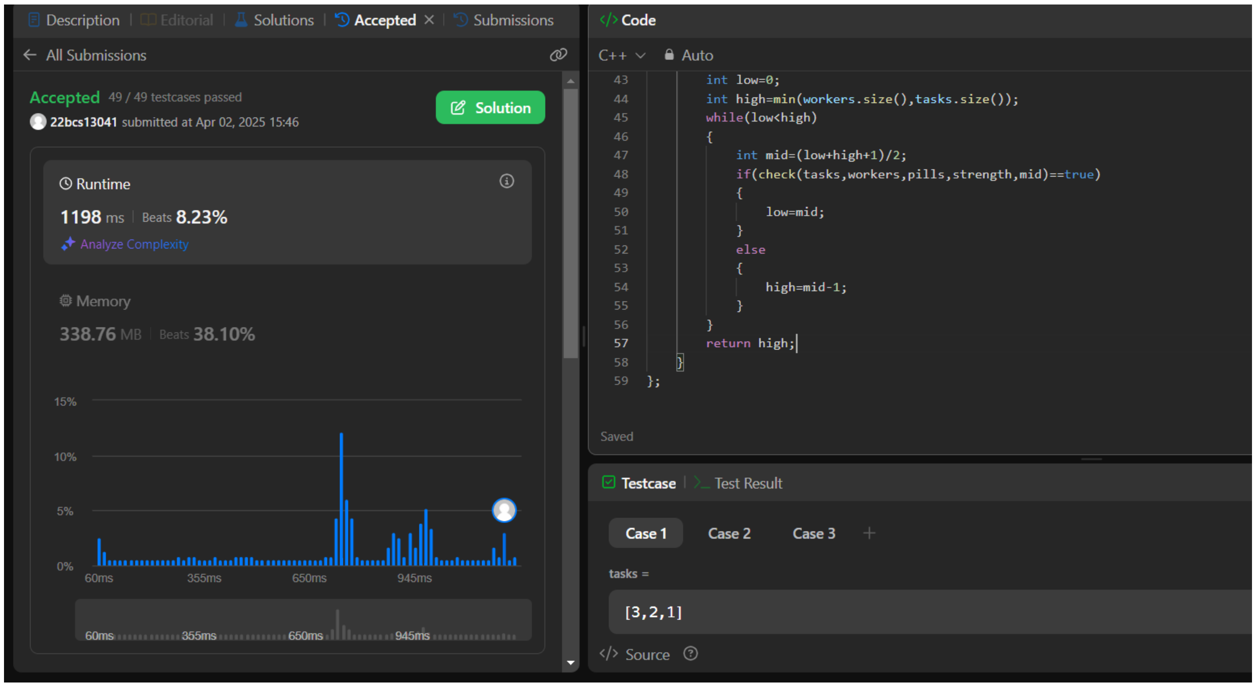
[5,1,3]

Max Number of Tasks You Can Assign

```
class Solution {
public:
    bool check(vector<int>& tasks, vector<int>& workers, int pills, int strength,int index)
    {
        multiset<int> st;
        for(auto it:workers)
        {
            st.insert(it);
        }
        for(int i=index-1;i>=0;i--)
        {
            auto it=st.lower_bound(tasks[i]);
            if(it!=st.end())
            {
                st.erase(it);
            }
            else
            {
                if(pills<=0)
                {
```

```
                return false;
            }
            else
            {
                it=st.lower_bound(tasks[i]-strength);
                if(it!=st.end())
                {
                    st.erase(it);
                    pills--;
                }
                else
                {
                    return false;
                }
            }
        }
    }
    return true;
}
int maxTaskAssign(vector<int>& tasks, vector<int>& workers, int pills, int strength) {
    sort(tasks.begin(),tasks.end());
    sort(workers.begin(),workers.end());
    int low=0;
    int high=min(workers.size(),tasks.size());
    while(low<high)
    {
        int mid=(low+high+1)/2;
        if(check(tasks,workers,pills,strength,mid)==true)
        {
            low=mid;
        }
        else
        {
            high=mid-1;
        }
    }
    return high;
}
```

};



Accepted 49 / 49 testcases passed

22bcs13041 submitted at Apr 02, 2025 15:46

Runtime

1198 ms | Beats 8.23%

Memory

338.76 MB | Beats 38.10%

```
        int low=0;
        int high=min(workers.size(),tasks.size());
        while(low<high)
        {
            int mid=(low+high+1)/2;
            if(check(tasks,workers,pills,strength,mid)==true)
            {
                low=mid;
            }
            else
            {
                high=mid-1;
            }
        }
        return high;
    }
};
```

Saved

tasks =

[3,2,1]