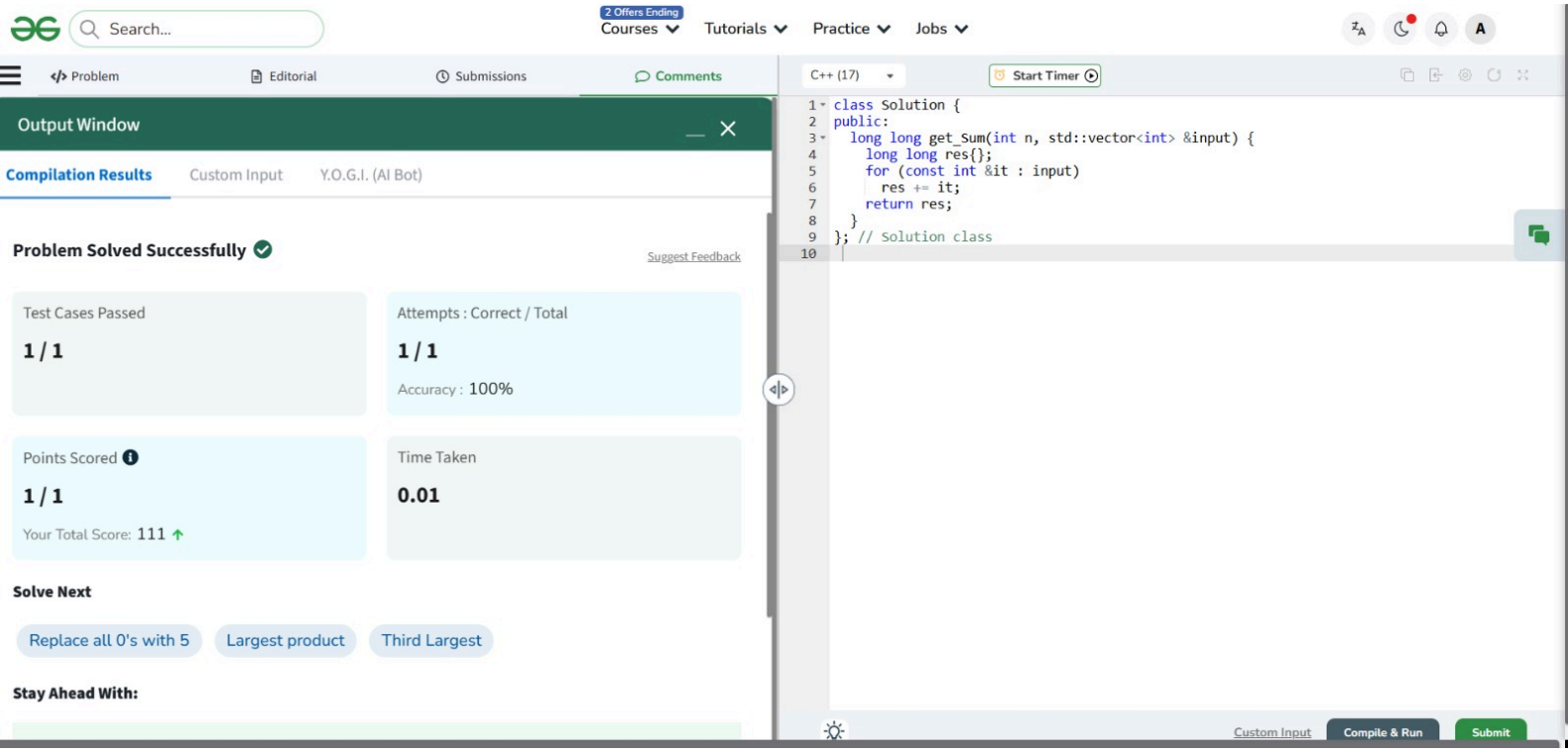
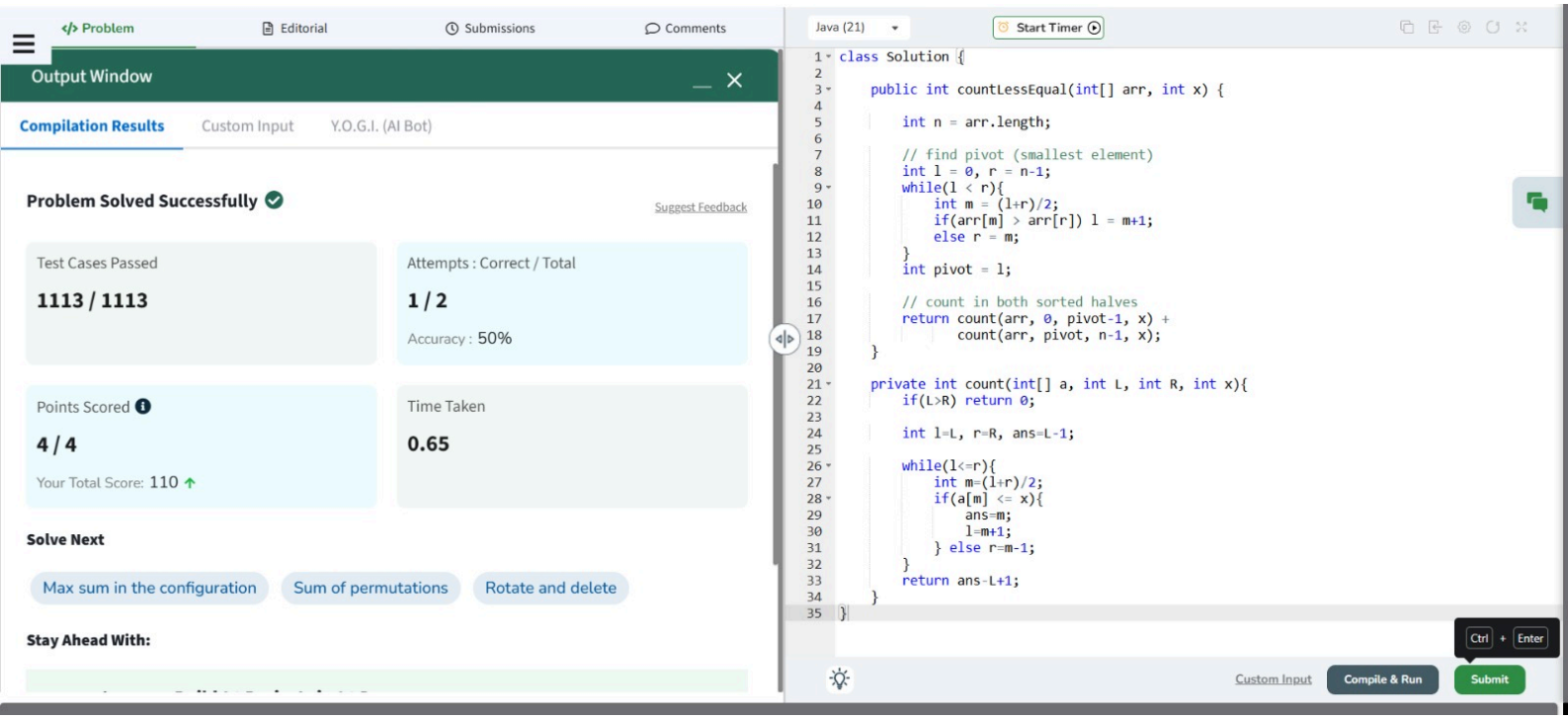
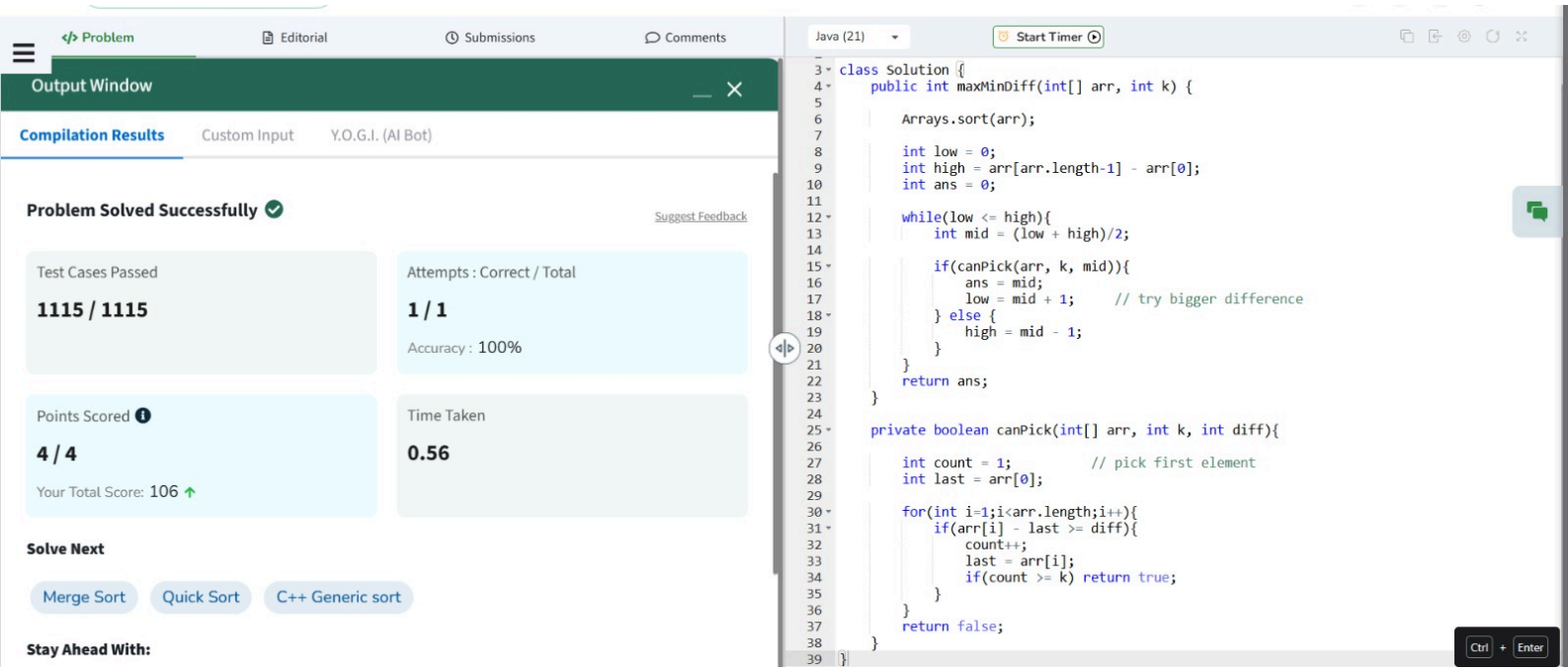
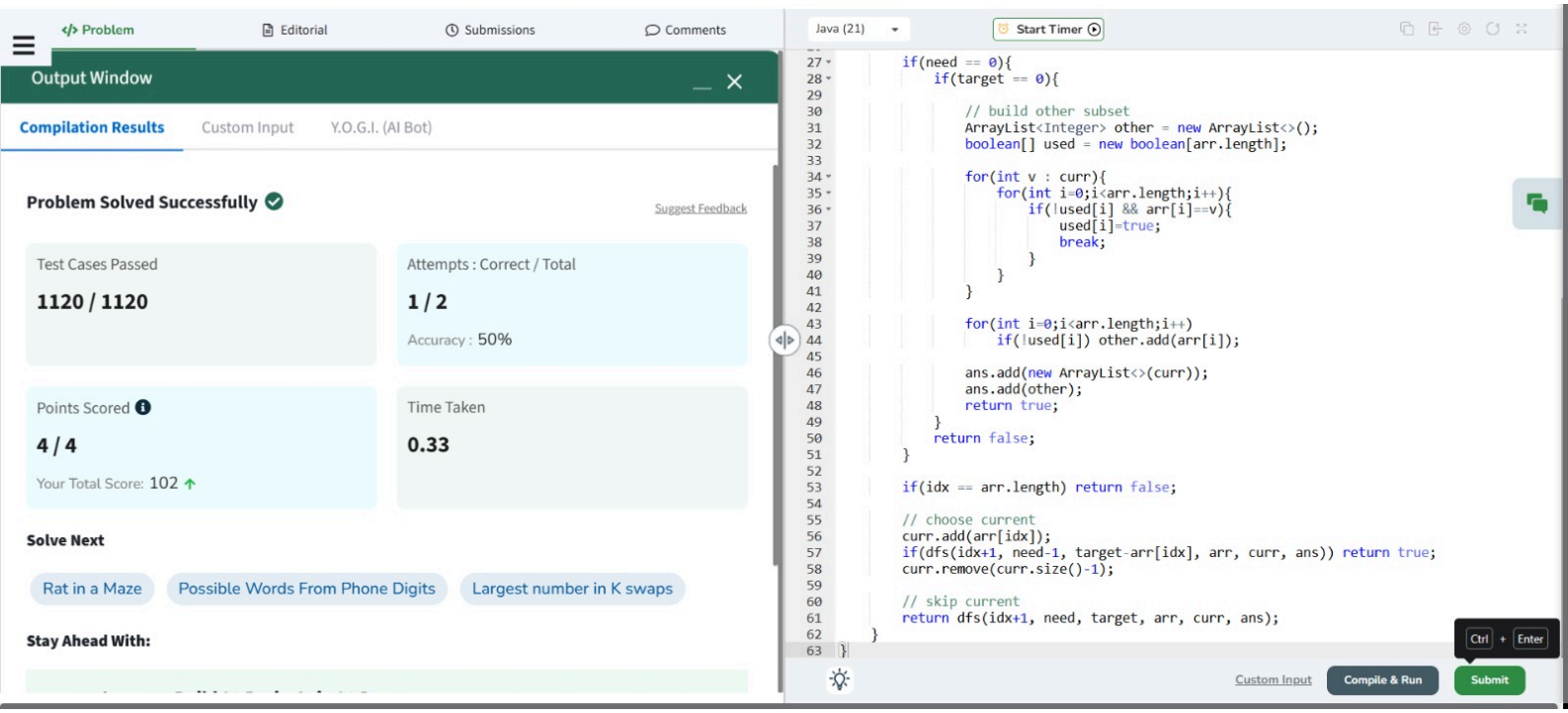
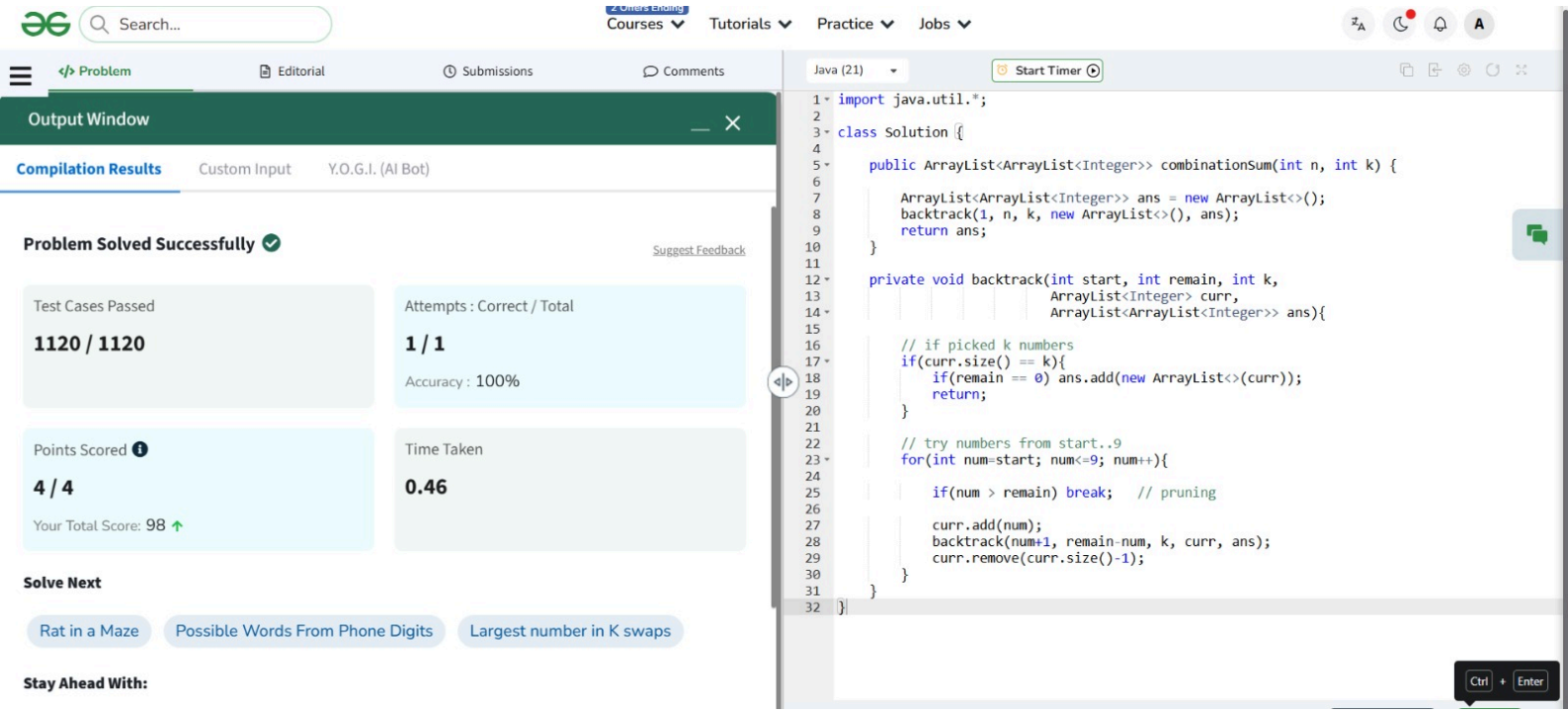
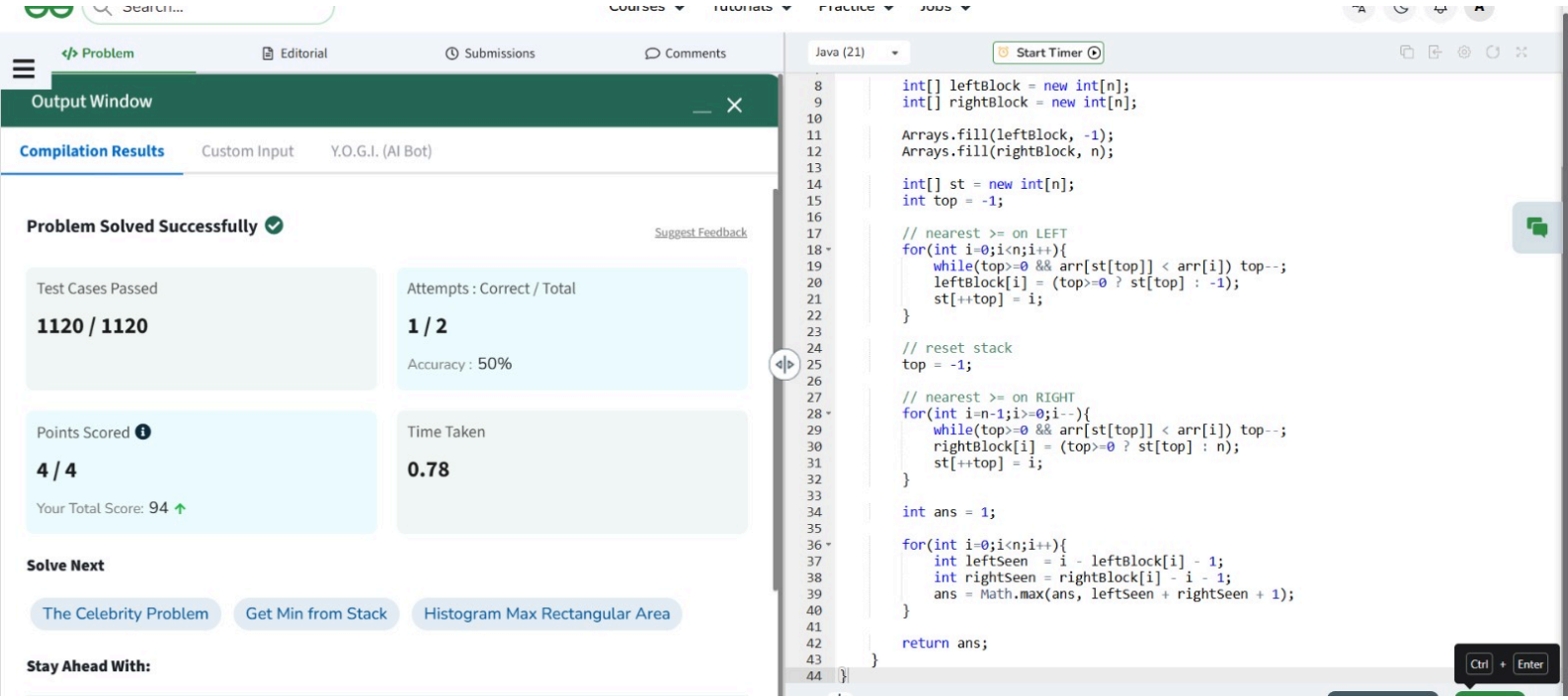

Search...
2 Offers Ending
Courses
Tutorials
Practice
Jobs
A
Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1 / 1
Attempts : Correct / Total
1 / 1
Accuracy : 100%
Points Scored
1 / 1
Your Total Score: 111
Time Taken
0.01
Solve Next
Replace all 0's with 5
Largest product
Third Largest
Stay Ahead With:
C++ (17)
Start Timer
1 class Solution {
2 public:
3 long long get_Sum(int n, std::vector<int> &input) {
4 long long res{};
5 for (const int &it : input)
6 res += it;
7 return res;
8 }
9 }; // Solution class
10
Custom Input
Compile & Run
Submit

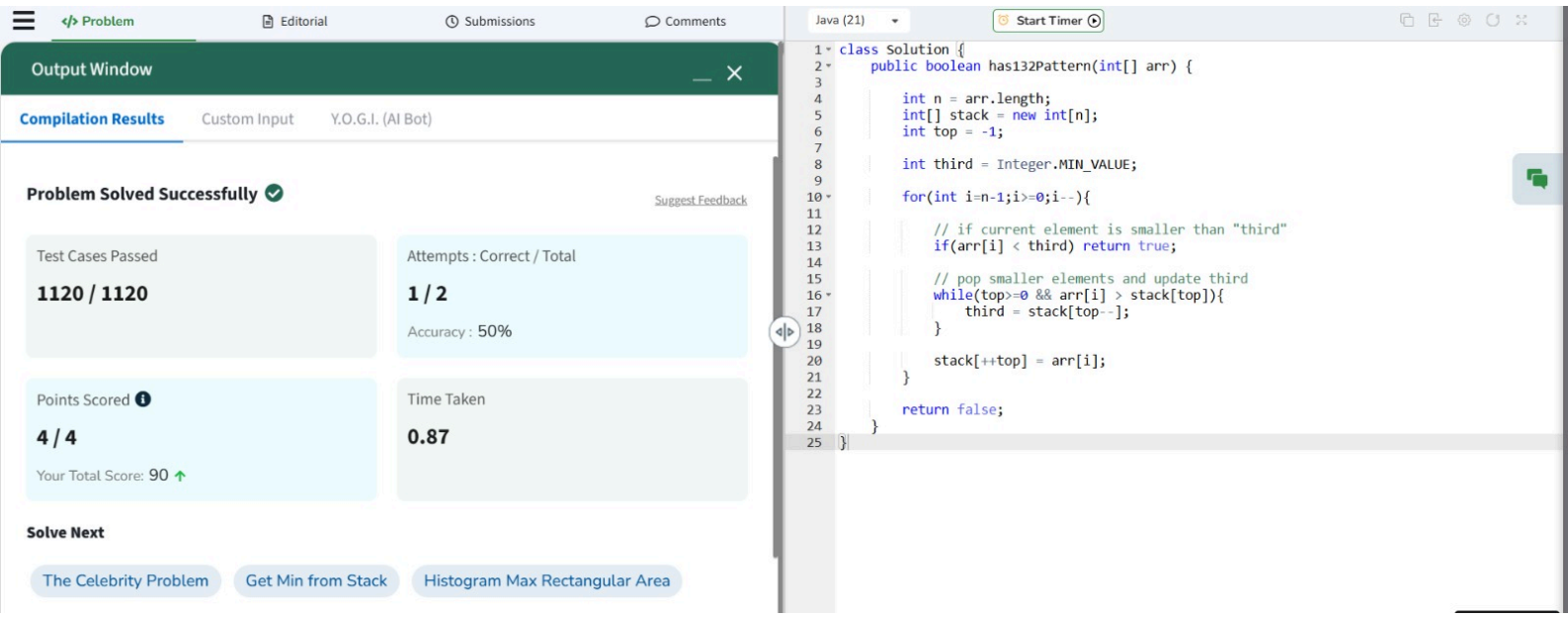

Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1113 / 1113
Attempts : Correct / Total
1 / 2
Accuracy : 50%
Points Scored
4 / 4
Your Total Score: 110
Time Taken
0.65
Solve Next
Max sum in the configuration
Sum of permutations
Rotate and delete
Stay Ahead With:
Java (21)
Start Timer
class Solution {
public int countLessEqual(int[] arr, int x) {
int n = arr.length;
// find pivot (smallest element)
int l = 0, r = n-1;
while(l < r){
int m = (l+r)/2;
if(arr[m] > arr[r]) l = m+1;
else r = m;
}
int pivot = l;
// count in both sorted halves
return count(arr, 0, pivot-1, x) +
count(arr, pivot, n-1, x);
}
private int count(int[] a, int L, int R, int x){
if(L>R) return 0;
int l=L, r=R, ans=L-1;
while(l<=r){
int m=(l+r)/2;
if(a[m] <= x){
ans=m;
l=m+1;
} else r=m-1;
}
return ans-L+1;
}
}
Ctrl + Enter
Custom Input
Compile & Run
Submit

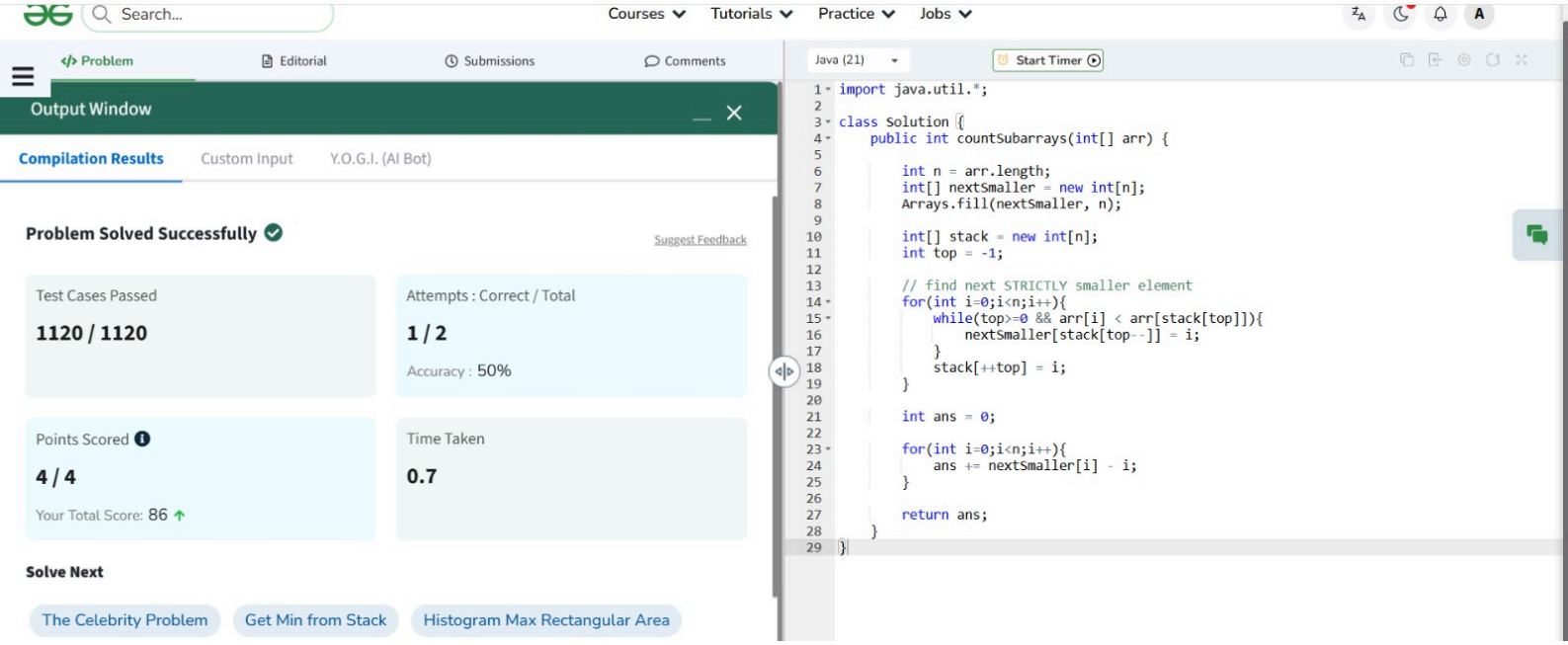

Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1115 / 1115
Attempts : Correct / Total
1 / 1
Accuracy : 100%
Points Scored
4 / 4
Your Total Score: 106
Time Taken
0.56
Solve Next
Merge Sort
Quick Sort
C++ Generic sort
Stay Ahead With:
Java (21)
Start Timer
class Solution {
public int maxMinDiff(int[] arr, int k) {
Arrays.sort(arr);
int low = 0;
int high = arr[arr.length-1] - arr[0];
int ans = 0;
while(low <= high){
int mid = (low + high)/2;
if(canPick(arr, k, mid)){
ans = mid;
low = mid + 1; // try bigger difference
} else {
high = mid - 1;
}
}
return ans;
}
private boolean canPick(int[] arr, int k, int diff){
int count = 1; // pick first element
int last = arr[0];
for(int i=1;i<arr.length;i++){
if(arr[i] - last >= diff){
count++;
last = arr[i];
if(count >= k) return true;
}
}
return false;
}
}
Ctrl + Enter

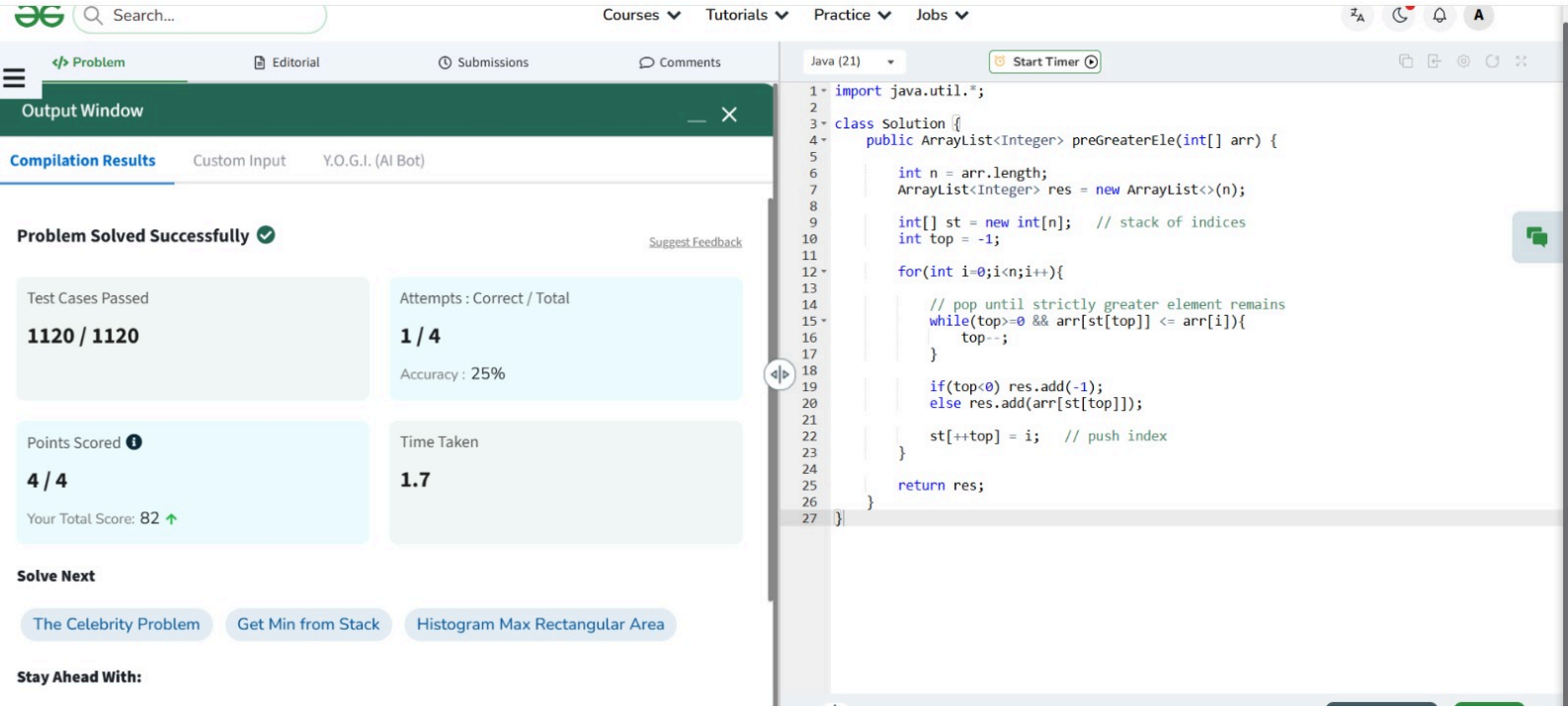

Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1120 / 1120
Attempts : Correct / Total
1 / 2
Accuracy : 50%
Points Scored
4 / 4
Your Total Score: 102
Time Taken
0.33
Solve Next
Rat in a Maze
Possible Words From Phone Digits
Largest number in K swaps
Stay Ahead With:
Java (21)
Start Timer
if(need == 0){
if(target == 0){
// build other subset
ArrayList<Integer> other = new ArrayList<>();
boolean[] used = new boolean[arr.length];
for(int v : curr){
for(int i=0;i<arr.length;i++){
if(!used[i] && arr[i]==v){
used[i]=true;
break;
}
}
}
for(int i=0;i<arr.length;i++)
if(!used[i]) other.add(arr[i]);
ans.add(new ArrayList<>(curr));
ans.add(other);
return true;
}
return false;
}
if(idx == arr.length) return false;
// choose current
curr.add(arr[idx]);
if(dfs(idx+1, need-1, target-arr[idx], arr, curr, ans)) return true;
curr.remove(curr.size()-1);
// skip current
return dfs(idx+1, need, target, arr, curr, ans);
}
}
Ctrl + Enter
Custom Input
Compile & Run
Submit

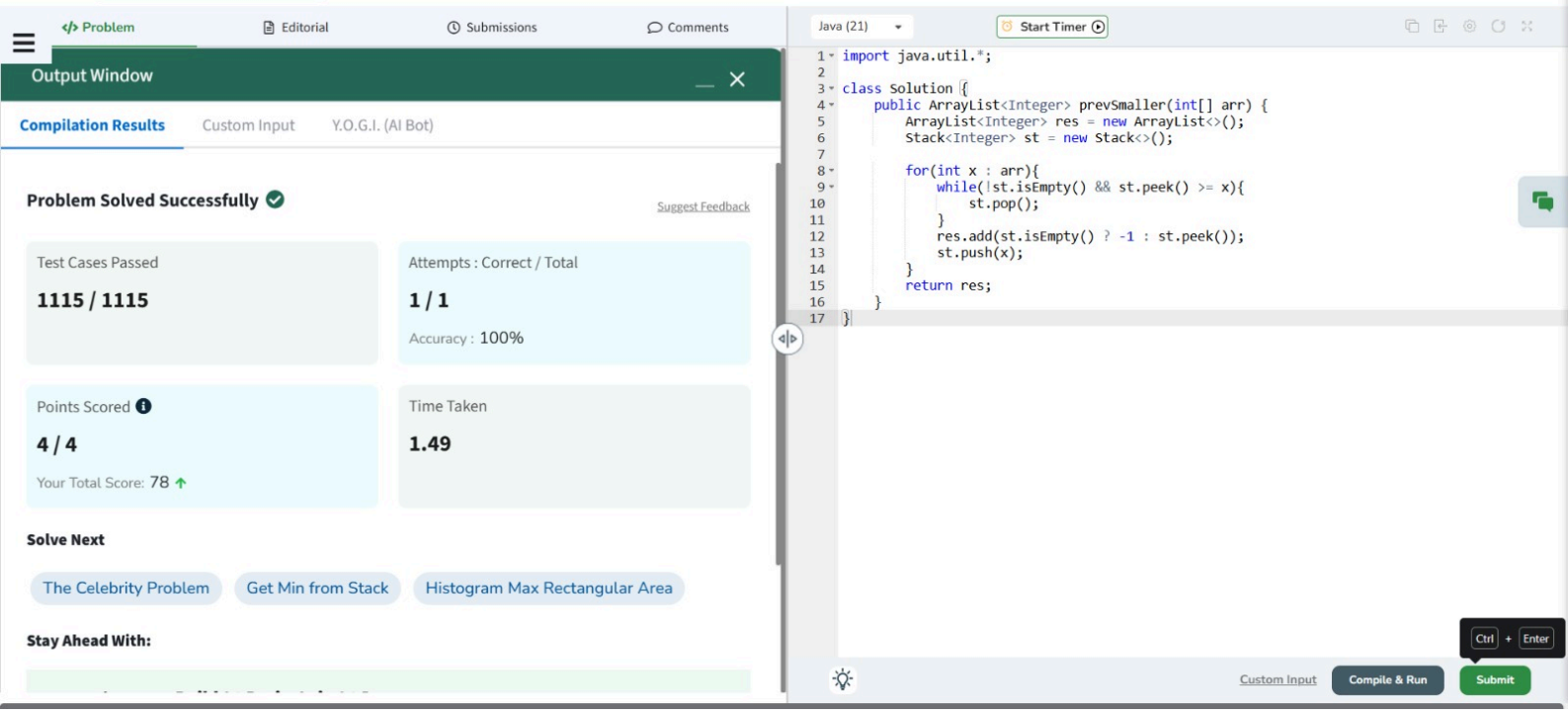

Search...
Courses
Tutorials
Practice
Jobs
Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1120 / 1120
Attempts : Correct / Total
1 / 1
Accuracy : 100%
Points Scored
4 / 4
Your Total Score: 98
Time Taken
0.46
Solve Next
Rat in a Maze
Possible Words From Phone Digits
Largest number in K swaps
Stay Ahead With:
Java (21)
Start Timer
import java.util.*;
class Solution {
public ArrayList<ArrayList<Integer>> combinationSum(int n, int k) {
ArrayList<ArrayList<Integer>> ans = new ArrayList<>();
backtrack(1, n, k, new ArrayList<>(), ans);
return ans;
}
private void backtrack(int start, int remain, int k,
ArrayList<Integer> curr,
ArrayList<ArrayList<Integer>> ans){
// if picked k numbers
if(curr.size() == k){
if(remain == 0) ans.add(new ArrayList<>(curr));
return;
}
// try numbers from start..9
for(int num=start; num<=9; num++){
if(num > remain) break; // pruning
curr.add(num);
backtrack(num+1, remain-num, k, curr, ans);
curr.remove(curr.size()-1);
}
}
}
Ctrl + Enter

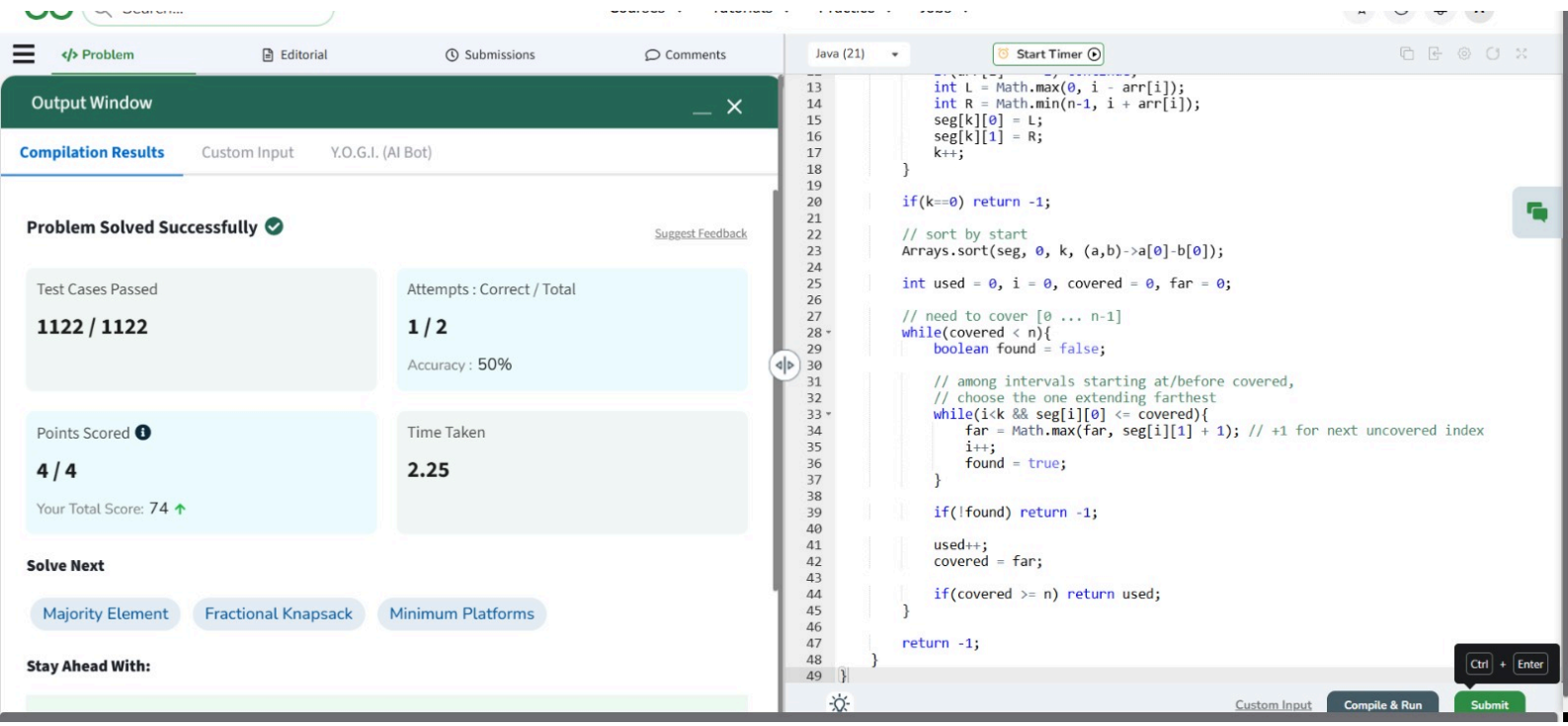

Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1120 / 1120
Attempts : Correct / Total
1 / 2
Accuracy : 50%
Points Scored
4 / 4
Your Total Score: 94
Time Taken
0.78
Solve Next
The Celebrity Problem
Get Min from Stack
Histogram Max Rectangular Area
Stay Ahead With:
Java (21)
Start Timer
int[] leftBlock = new int[n];
int[] rightBlock = new int[n];
Arrays.fill(leftBlock, -1);
Arrays.fill(rightBlock, n);
int[] st = new int[n];
int top = -1;
// nearest >= on LEFT
for(int i=0;i<n;i++){
while(top>=0 && arr[st[top]] < arr[i]) top--;
leftBlock[i] = (top>=0 ? st[top] : -1);
st[++top] = i;
}
// reset stack
top = -1;
// nearest >= on RIGHT
for(int i=n-1;i>=0;i--){
while(top>=0 && arr[st[top]] < arr[i]) top--;
rightBlock[i] = (top>=0 ? st[top] : n);
st[++top] = i;
}
int ans = 1;
for(int i=0;i<n;i++){
int leftSeen = i - leftBlock[i] - 1;
int rightSeen = rightBlock[i] - i - 1;
ans = Math.max(ans, leftSeen + rightSeen + 1);
}
return ans;
}
}
Ctrl + Enter

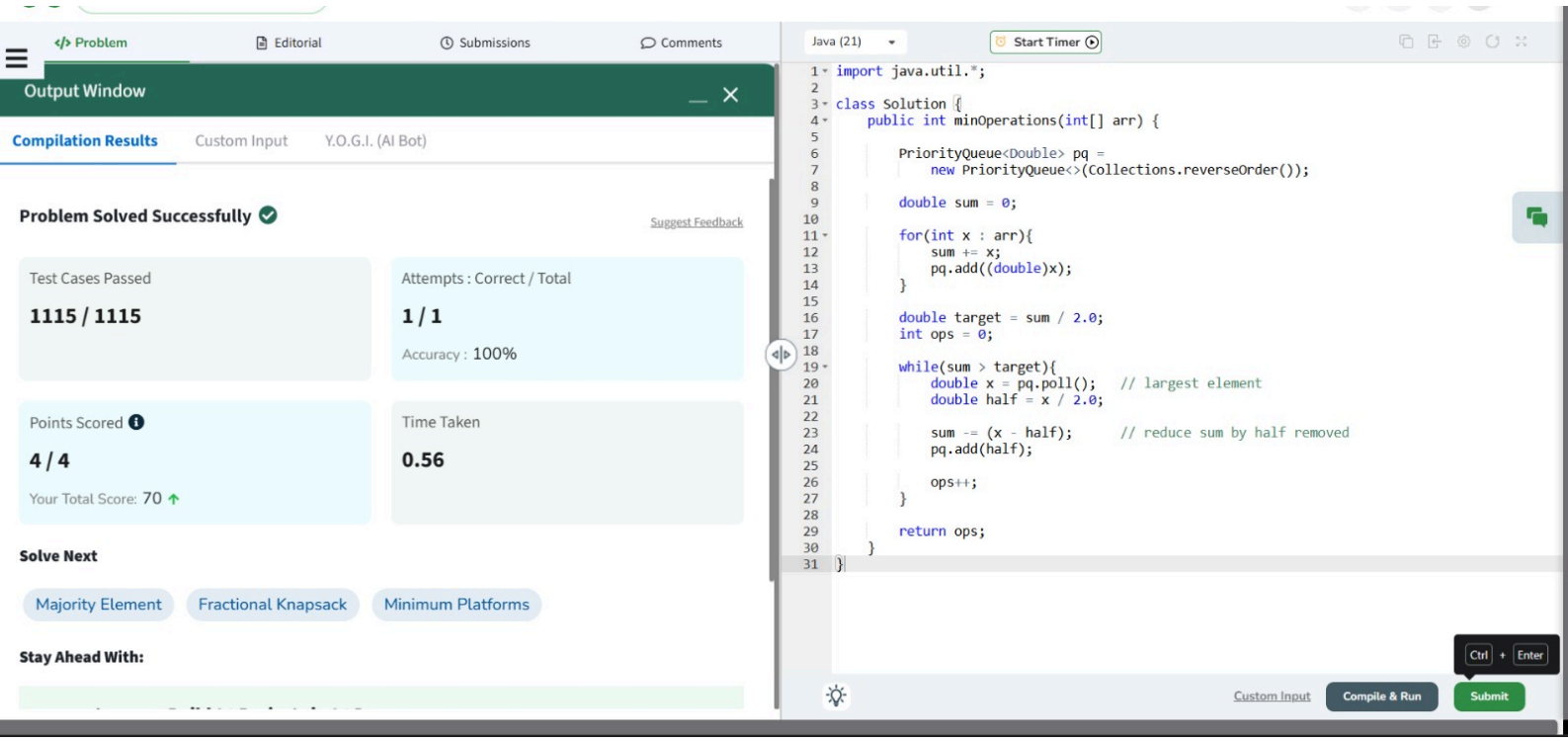

Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1120 / 1120
Attempts : Correct / Total
1 / 2
Accuracy : 50%
Points Scored
4 / 4
Your Total Score: 90
Time Taken
0.87
Solve Next
The Celebrity Problem
Get Min from Stack
Histogram Max Rectangular Area
Java (21)
Start Timer
class Solution {
public boolean has132Pattern(int[] arr) {
int n = arr.length;
int[] stack = new int[n];
int top = -1;
int third = Integer.MIN_VALUE;
for(int i=n-1;i>=0;i--){
// if current element is smaller than "third"
if(arr[i] < third) return true;
// pop smaller elements and update third
while(top>=0 && arr[i] > stack[top]){
third = stack[top--];
}
stack[++top] = arr[i];
}
return false;
}
}

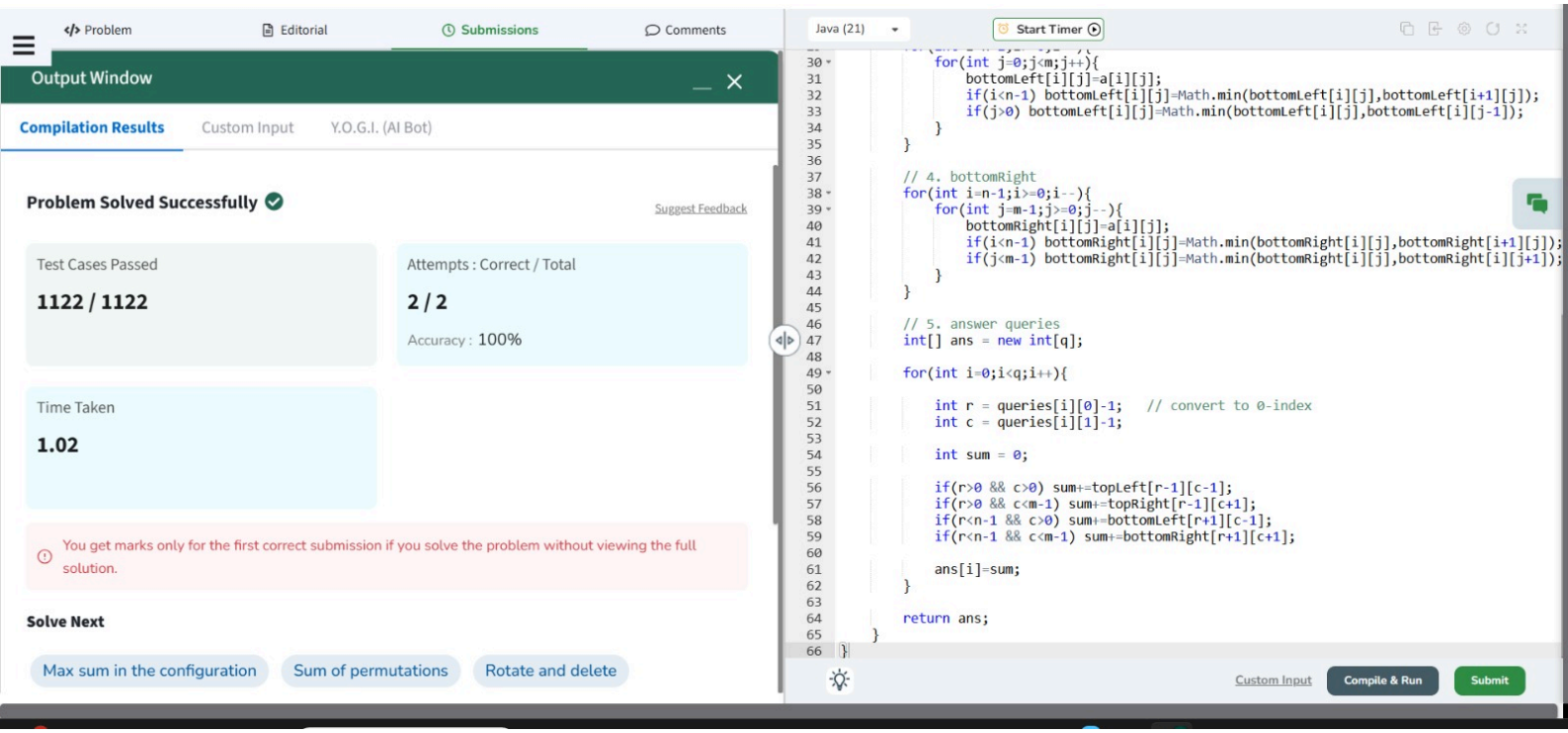

Search...
Courses
Tutorials
Practice
Jobs
Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1120 / 1120
Attempts : Correct / Total
1 / 2
Accuracy : 50%
Points Scored
4 / 4
Your Total Score: 86
Time Taken
0.7
Solve Next
The Celebrity Problem
Get Min from Stack
Histogram Max Rectangular Area
Java (21)
Start Timer
import java.util.*;
class Solution {
public int countSubarrays(int[] arr) {
int n = arr.length;
int[] nextSmaller = new int[n];
Arrays.fill(nextSmaller, n);
int[] stack = new int[n];
int top = -1;
// find next STRICTLY smaller element
for(int i=0;i<n;i++){
while(top>=0 && arr[i] < arr[stack[top]]){
nextSmaller[stack[top--]] = i;
}
stack[++top] = i;
}
int ans = 0;
for(int i=0;i<n;i++){
ans += nextSmaller[i] - i;
}
return ans;
}
}


Search...
Courses
Tutorials
Practice
Jobs
Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1120 / 1120
Attempts : Correct / Total
1 / 4
Accuracy : 25%
Points Scored
4 / 4
Your Total Score: 82
Time Taken
1.7
Solve Next
The Celebrity Problem
Get Min from Stack
Histogram Max Rectangular Area
Stay Ahead With:
Java (21)
Start Timer
import java.util.*;
class Solution {
public ArrayList<Integer> preGreaterEle(int[] arr) {
int n = arr.length;
ArrayList<Integer> res = new ArrayList<>(n);
int[] st = new int[n]; // stack of indices
int top = -1;
for(int i=0;i<n;i++){
// pop until strictly greater element remains
while(top>=0 && arr[st[top]] <= arr[i]){
top--;
}
if(top<0) res.add(-1);
else res.add(arr[st[top]]);
st[++top] = i; // push index
}
return res;
}
}


Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1115 / 1115
Attempts : Correct / Total
1 / 1
Accuracy : 100%
Points Scored
4 / 4
Your Total Score: 78
Time Taken
1.49
Solve Next
The Celebrity Problem
Get Min from Stack
Histogram Max Rectangular Area
Stay Ahead With:
Java (21)
Start Timer
1 import java.util.*;
2
3 class Solution {
4 public ArrayList<Integer> prevSmaller(int[] arr) {
5 ArrayList<Integer> res = new ArrayList<>();
6 Stack<Integer> st = new Stack<>();
7
8 for(int x : arr){
9 while(!st.isEmpty() && st.peek() >= x){
10 st.pop();
11 }
12 res.add(st.isEmpty() ? -1 : st.peek());
13 st.push(x);
14 }
15 return res;
16 }
17 }
Ctrl + Enter
Custom Input
Compile & Run
Submit


Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1122 / 1122
Attempts : Correct / Total
1 / 2
Accuracy : 50%
Points Scored
4 / 4
Your Total Score: 74
Time Taken
2.25
Solve Next
Majority Element
Fractional Knapsack
Minimum Platforms
Stay Ahead With:
Java (21)
Start Timer
int L = Math.max(0, i - arr[i]);
int R = Math.min(n-1, i + arr[i]);
seg[k][0] = L;
seg[k][1] = R;
k++;
}
if(k==0) return -1;
// sort by start
Arrays.sort(seg, 0, k, (a,b)->a[0]-b[0]);
int used = 0, i = 0, covered = 0, far = 0;
// need to cover [0 ... n-1]
while(covered < n){
boolean found = false;
// among intervals starting at/before covered,
// choose the one extending farthest
while(i<k && seg[i][0] <= covered){
far = Math.max(far, seg[i][1] + 1); // +1 for next uncovered index
i++;
found = true;
}
if(!found) return -1;
used++;
covered = far;
if(covered >= n) return used;
}
return -1;
}
}
Ctrl + Enter
Custom Input
Compile & Run
Submit


Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1115 / 1115
Attempts : Correct / Total
1 / 1
Accuracy : 100%
Points Scored
4 / 4
Your Total Score: 70
Time Taken
0.56
Solve Next
Majority Element
Fractional Knapsack
Minimum Platforms
Stay Ahead With:
Java (21)
Start Timer
import java.util.*;
class Solution {
public int minOperations(int[] arr) {
PriorityQueue<Double> pq =
new PriorityQueue<>(Collections.reverseOrder());
double sum = 0;
for(int x : arr){
sum += x;
pq.add((double)x);
}
double target = sum / 2.0;
int ops = 0;
while(sum > target){
double x = pq.poll(); // largest element
double half = x / 2.0;
sum -= (x - half); // reduce sum by half removed
pq.add(half);
ops++;
}
return ops;
}
}
Ctrl + Enter
Custom Input
Compile & Run
Submit


Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1122 / 1122
Attempts : Correct / Total
2 / 2
Accuracy : 100%
Time Taken
1.02
You get marks only for the first correct submission if you solve the problem without viewing the full solution.
Solve Next
Max sum in the configuration
Sum of permutations
Rotate and delete
Java (21)
Start Timer
for(int j=0;j<m;j++){
bottomLeft[i][j]=a[i][j];
if(i<n-1) bottomLeft[i][j]=Math.min(bottomLeft[i][j],bottomLeft[i+1][j]);
if(j>0) bottomLeft[i][j]=Math.min(bottomLeft[i][j],bottomLeft[i][j-1]);
}
}
// 4. bottomRight
for(int i=n-1;i>=0;i--){
for(int j=m-1;j>=0;j--){
bottomRight[i][j]=a[i][j];
if(i<n-1) bottomRight[i][j]=Math.min(bottomRight[i][j],bottomRight[i+1][j]);
if(j<m-1) bottomRight[i][j]=Math.min(bottomRight[i][j],bottomRight[i][j+1]);
}
}
// 5. answer queries
int[] ans = new int[q];
for(int i=0;i<q;i++){
int r = queries[i][0]-1; // convert to 0-index
int c = queries[i][1]-1;
int sum = 0;
if(r>0 && c>0) sum+=topLeft[r-1][c-1];
if(r>0 && c<m-1) sum+=topRight[r-1][c+1];
if(r<n-1 && c>0) sum+=bottomLeft[r+1][c-1];
if(r<n-1 && c<m-1) sum+=bottomRight[r+1][c+1];
ans[i]=sum;
}
return ans;
}
}
Custom Input
Compile & Run
Submit

Problem | Editorial | Submissions | Comments

My Submissions

Refresh

Time (IST)	Status	Marks	Lang	Test Cases	Code
2026-02-20 11:30:02	Correct	4	java	214 / 214	View

Java (21) | Start Timer

```java
import java.util.*;

class Solution {

    static ArrayList<Long> submatrixSum(long[][] a, int n, int m, int[][] query, int q) {

        // Step 1: build prefix matrix
        long[][] pre = new long[n][m];

        for(int i=0;i<n;i++){
            for(int j=0;j<m;j++){
                pre[i][j]=a[i][j];

                if(i>0) pre[i][j]+=pre[i-1][j];
                if(j>0) pre[i][j]+=pre[i][j-1];
                if(i>0 && j>0) pre[i][j]-=pre[i-1][j-1];
            }
        }

        // Step 2: answer queries
        ArrayList<Long> ans = new ArrayList<>();

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

            int r1=query[k][0];
            int c1=query[k][1];
            int r2=query[k][2];
            int c2=query[k][3];

            long sum = pre[r2][c2];

            if(r1>0) sum-=pre[r1-1][c2];
            if(c1>0) sum-=pre[r2][c1-1];
            if(r1>0 && c1>0) sum+=pre[r1-1][c1-1];

            ans.add(sum);
        }
```