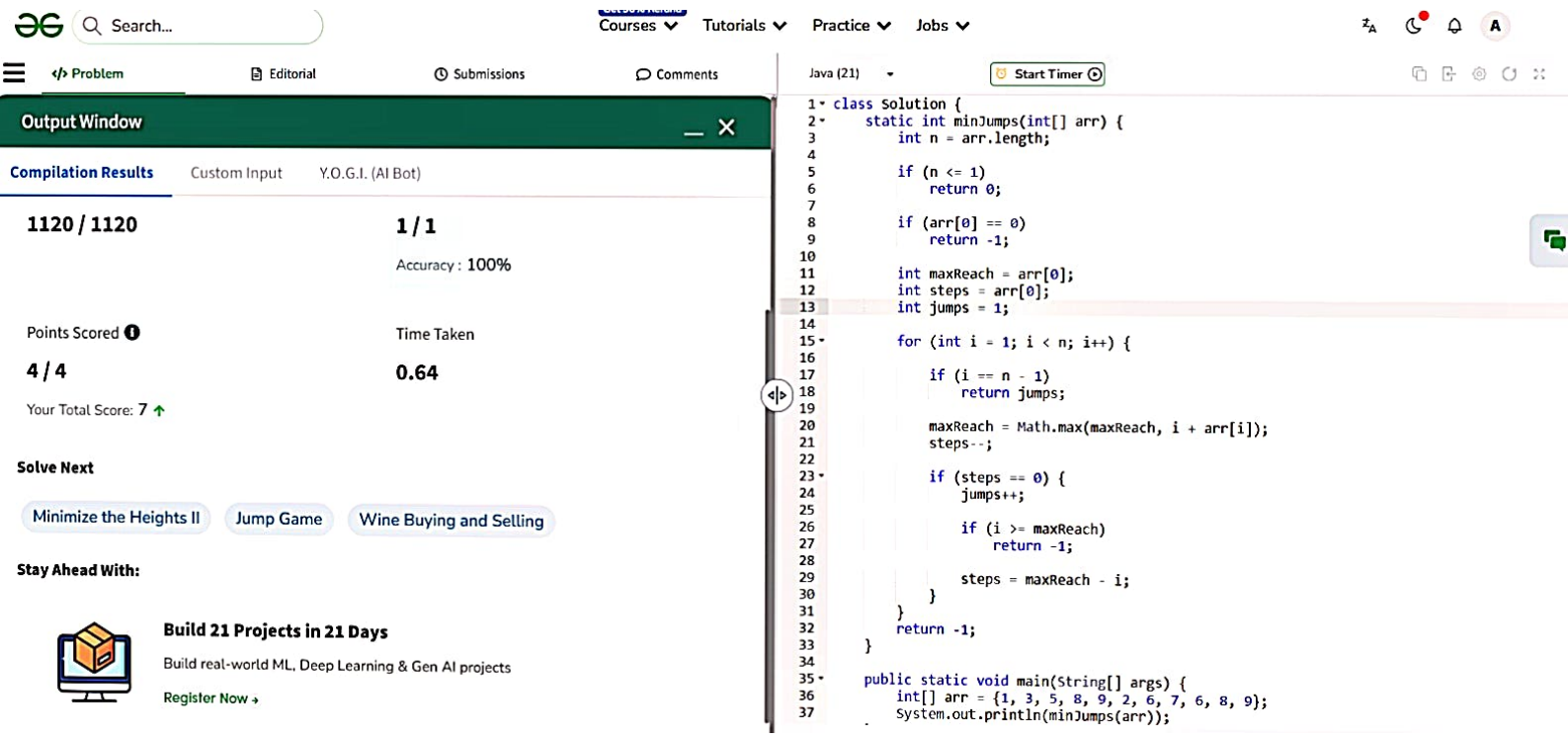
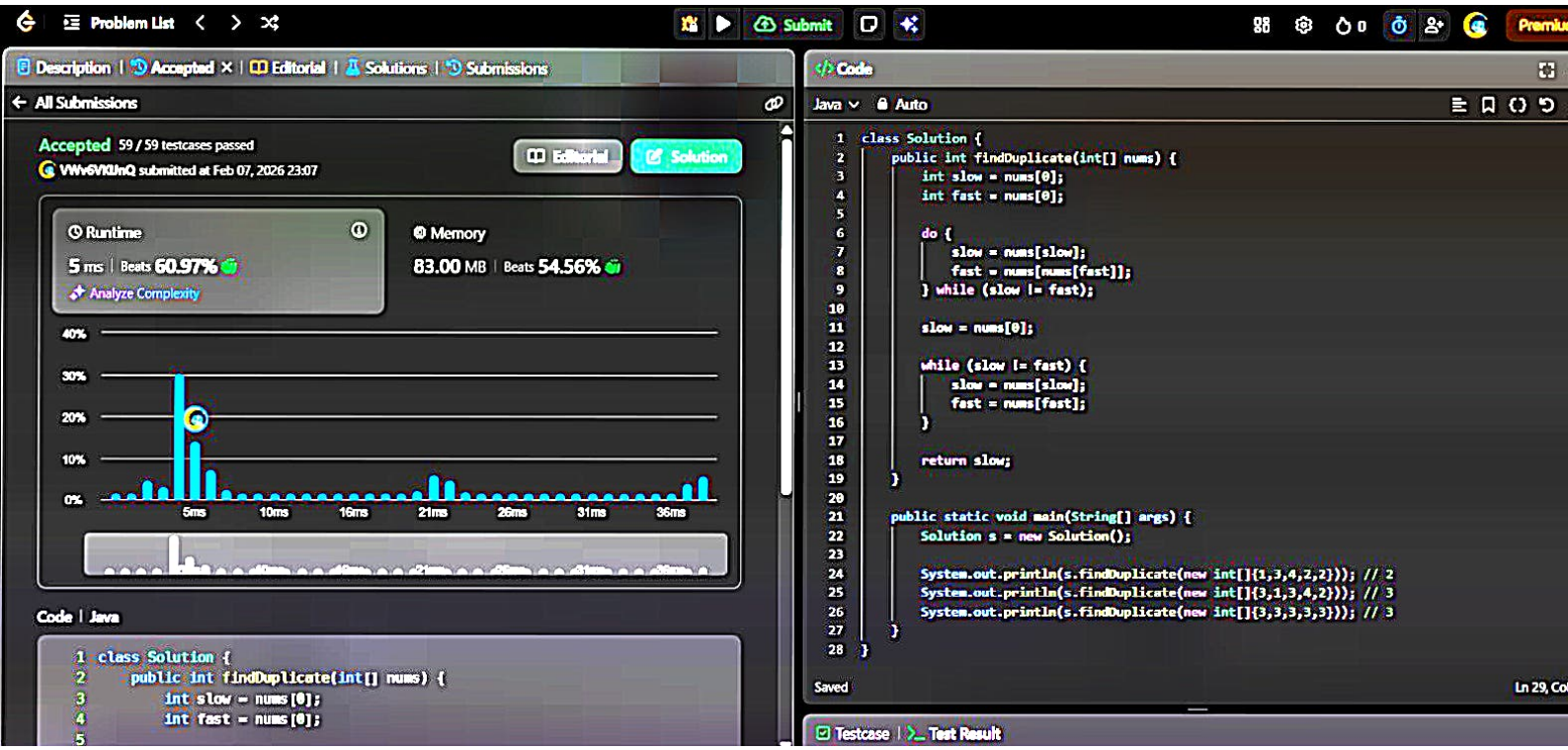
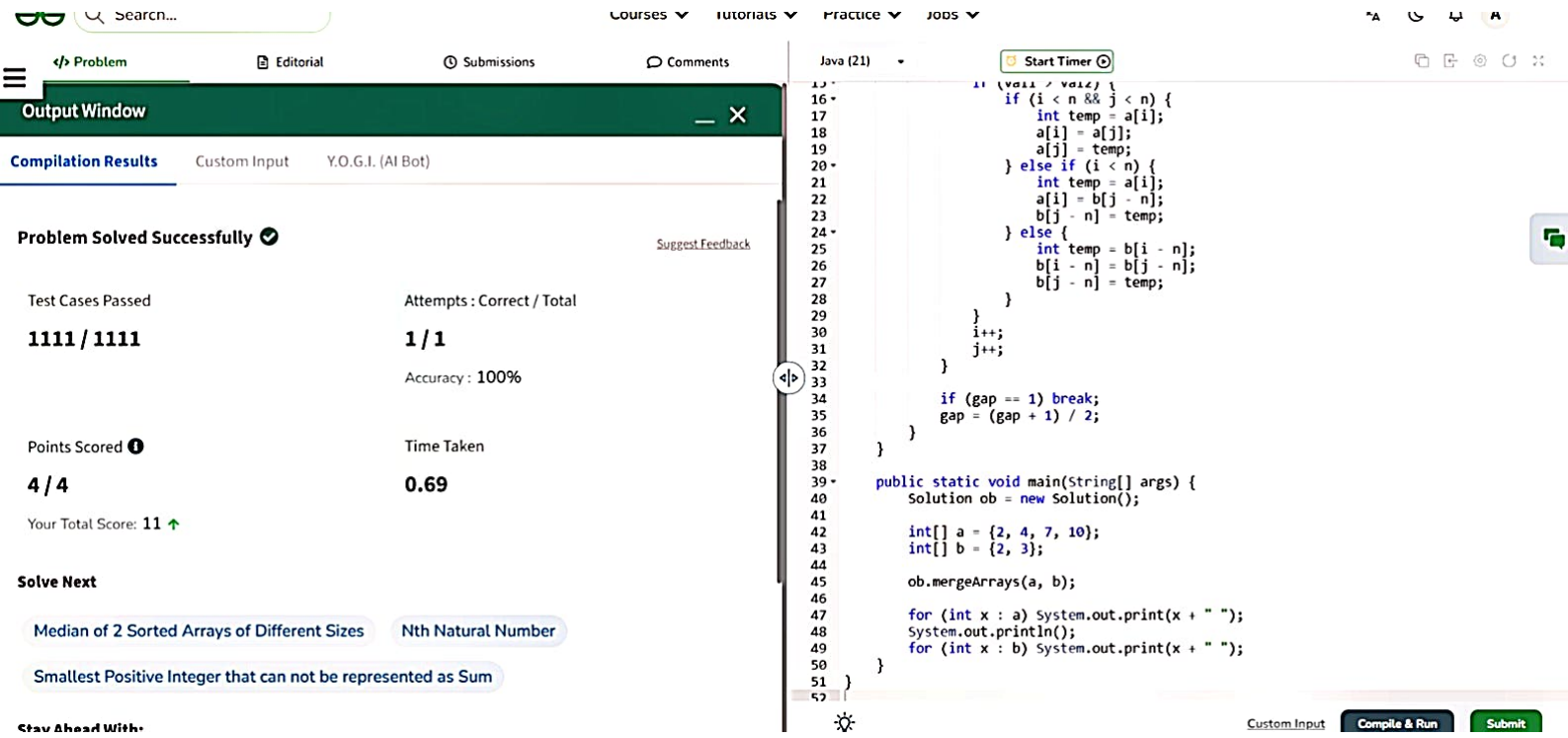
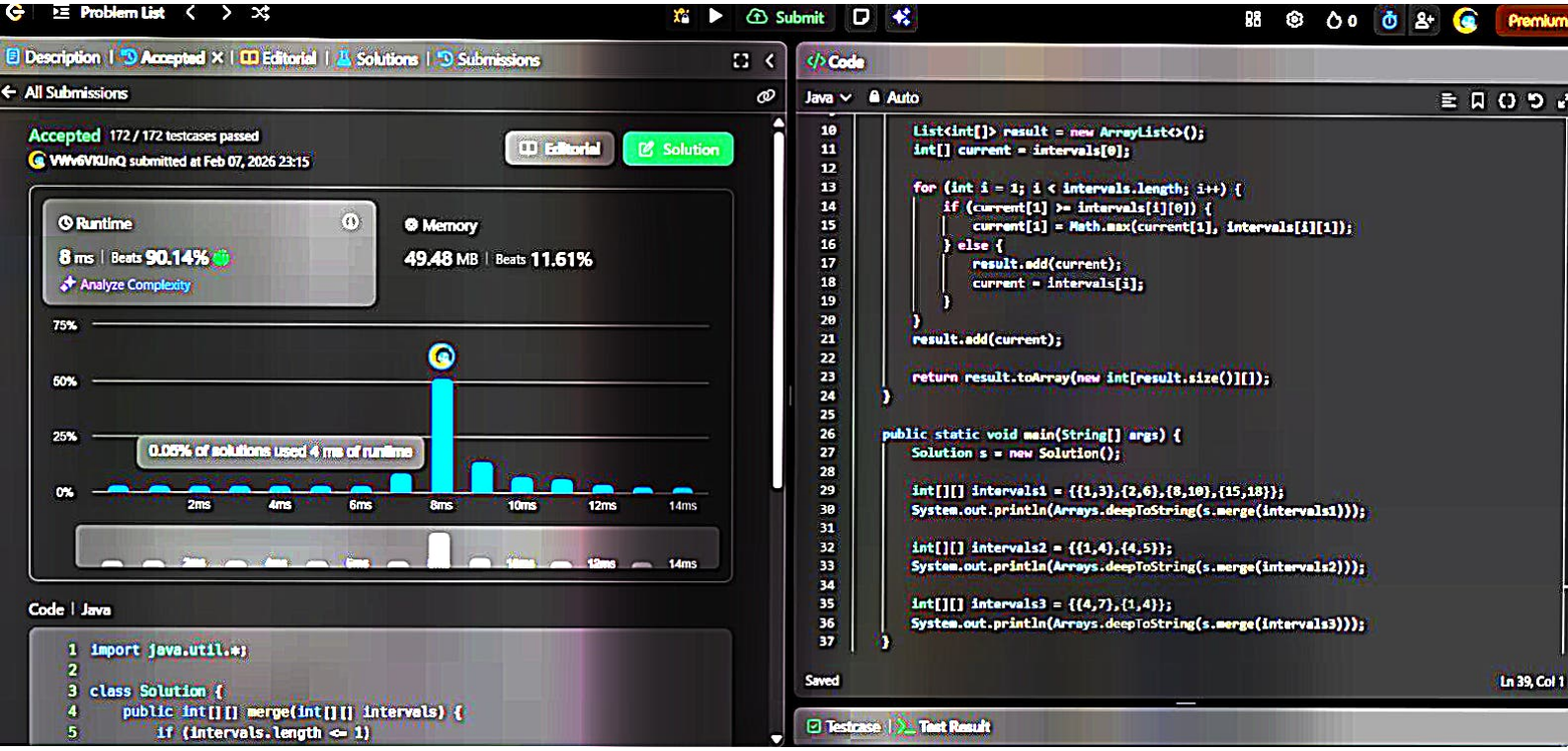
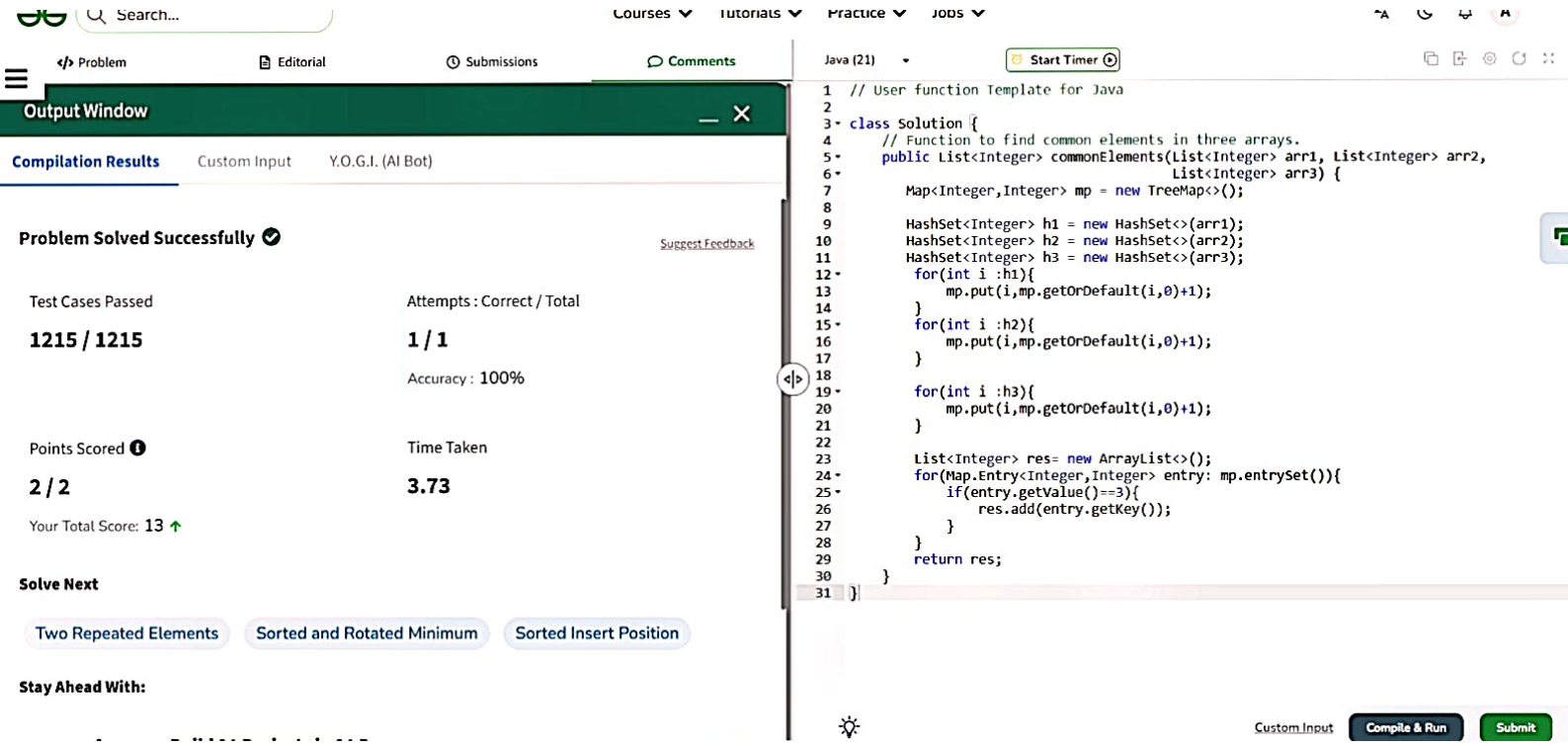
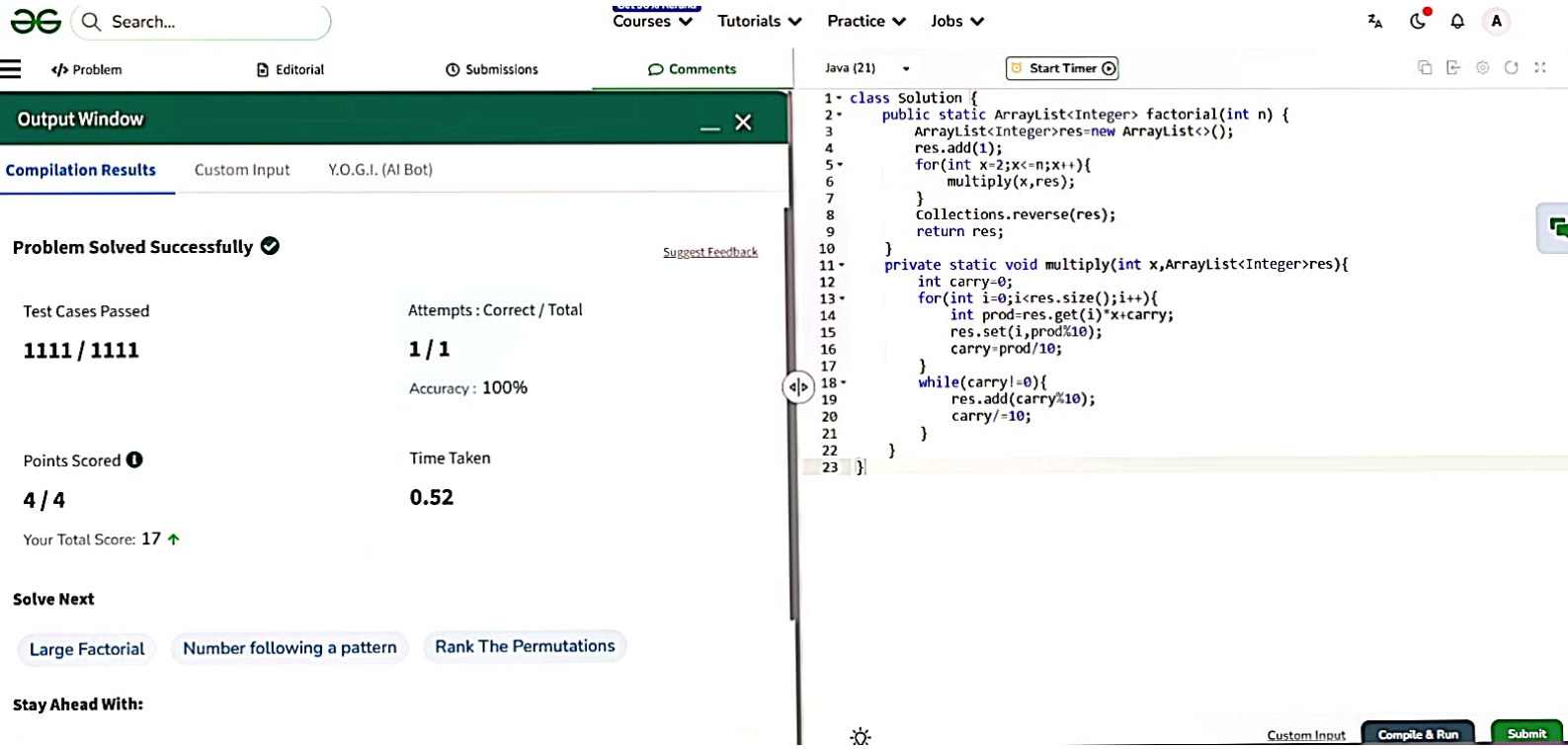

Search...
Courses
Tutorials
Practice
Jobs
Problem
Editorial
Submissions
Comments
Java (21)
Start Timer
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
1120 / 1120
1 / 1
Accuracy : 100%
Points Scored
Time Taken
4 / 4
0.64
Your Total Score: 7
Solve Next
Minimize the Heights II
Jump Game
Wine Buying and Selling
Stay Ahead With:
Build 21 Projects in 21 Days
Build real-world ML, Deep Learning & Gen AI projects
Register Now
class Solution {
static int minJumps(int[] arr) {
int n = arr.length;
if (n <= 1)
return 0;
if (arr[0] == 0)
return -1;
int maxReach = arr[0];
int steps = arr[0];
int jumps = 1;
for (int i = 1; i < n; i++) {
if (i == n - 1)
return jumps;
maxReach = Math.max(maxReach, i + arr[i]);
steps--;
if (steps == 0) {
jumps++;
if (i >= maxReach)
return -1;
steps = maxReach - i;
}
}
return -1;
}
public static void main(String[] args) {
int[] arr = {1, 3, 5, 8, 9, 2, 6, 7, 6, 8, 9};
System.out.println(minJumps(arr));

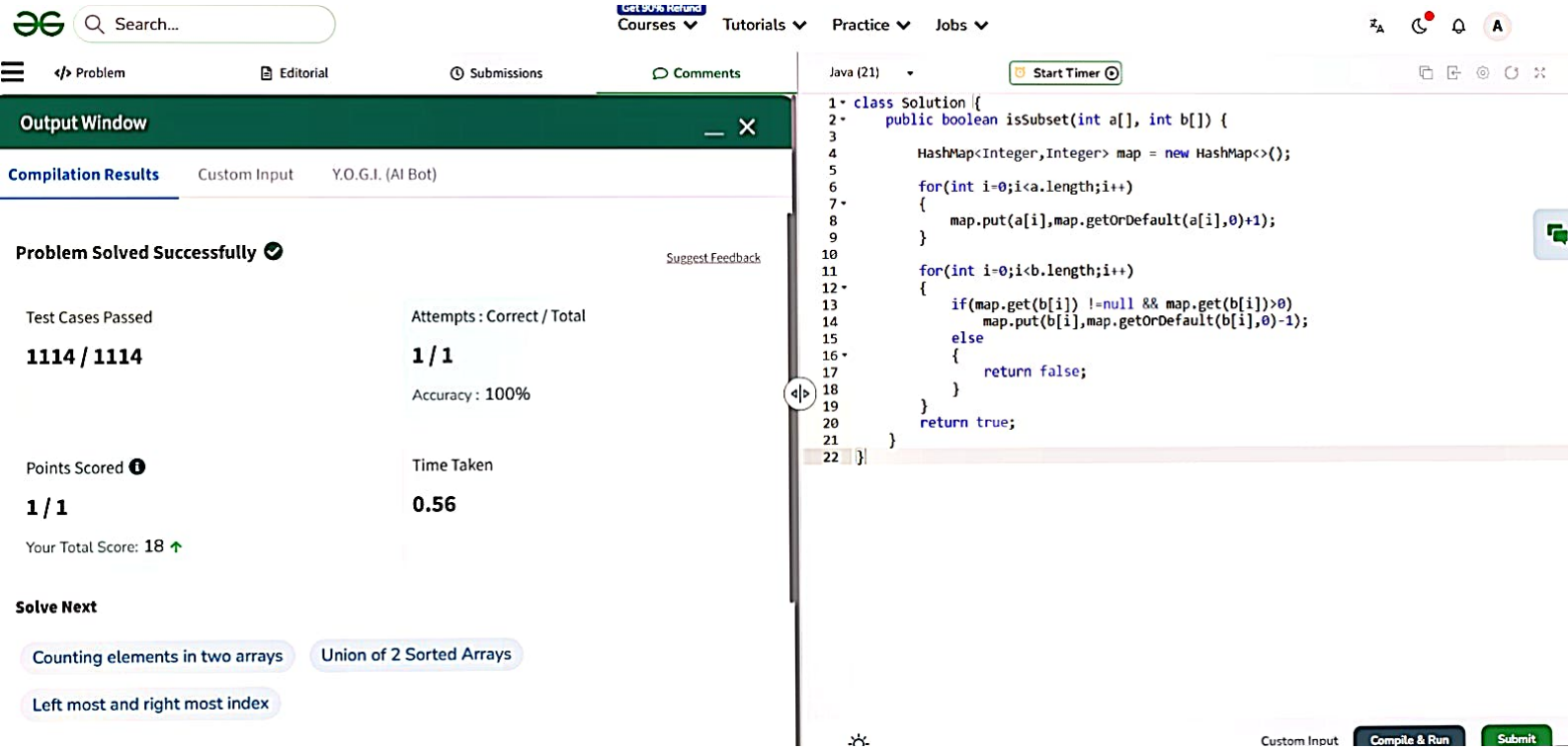

Problem List
Submit
Premiu
Description | Accepted | Editorial | Solutions | Submissions
All Submissions
Accepted 59 / 59 testcases passed
VWv6VKUnQ submitted at Feb 07, 2026 23:07
Editorial
Solution
Runtime
5 ms | Beats 60.97%
Analyze Complexity
Memory
83.00 MB | Beats 54.56%
40%
30%
20%
10%
0%
5ms
10ms
16ms
21ms
26ms
31ms
36ms
Code | Java
1 class Solution {
2 public int findDuplicate(int[] nums) {
3 int slow = nums[0];
4 int fast = nums[0];
5
Code
Java
Auto
1 class Solution {
2 public int findDuplicate(int[] nums) {
3 int slow = nums[0];
4 int fast = nums[0];
5
6 do {
7 slow = nums[slow];
8 fast = nums[nums[fast]];
9 } while (slow != fast);
10
11 slow = nums[0];
12
13 while (slow != fast) {
14 slow = nums[slow];
15 fast = nums[fast];
16 }
17
18 return slow;
19 }
20
21 public static void main(String[] args) {
22 Solution s = new Solution();
23
24 System.out.println(s.findDuplicate(new int[]{1,3,4,2,2})); // 2
25 System.out.println(s.findDuplicate(new int[]{3,1,3,4,2})); // 3
26 System.out.println(s.findDuplicate(new int[]{3,3,3,3,3})); // 3
27 }
28 }
Saved
Ln 29, Co
Testcase | Test Result

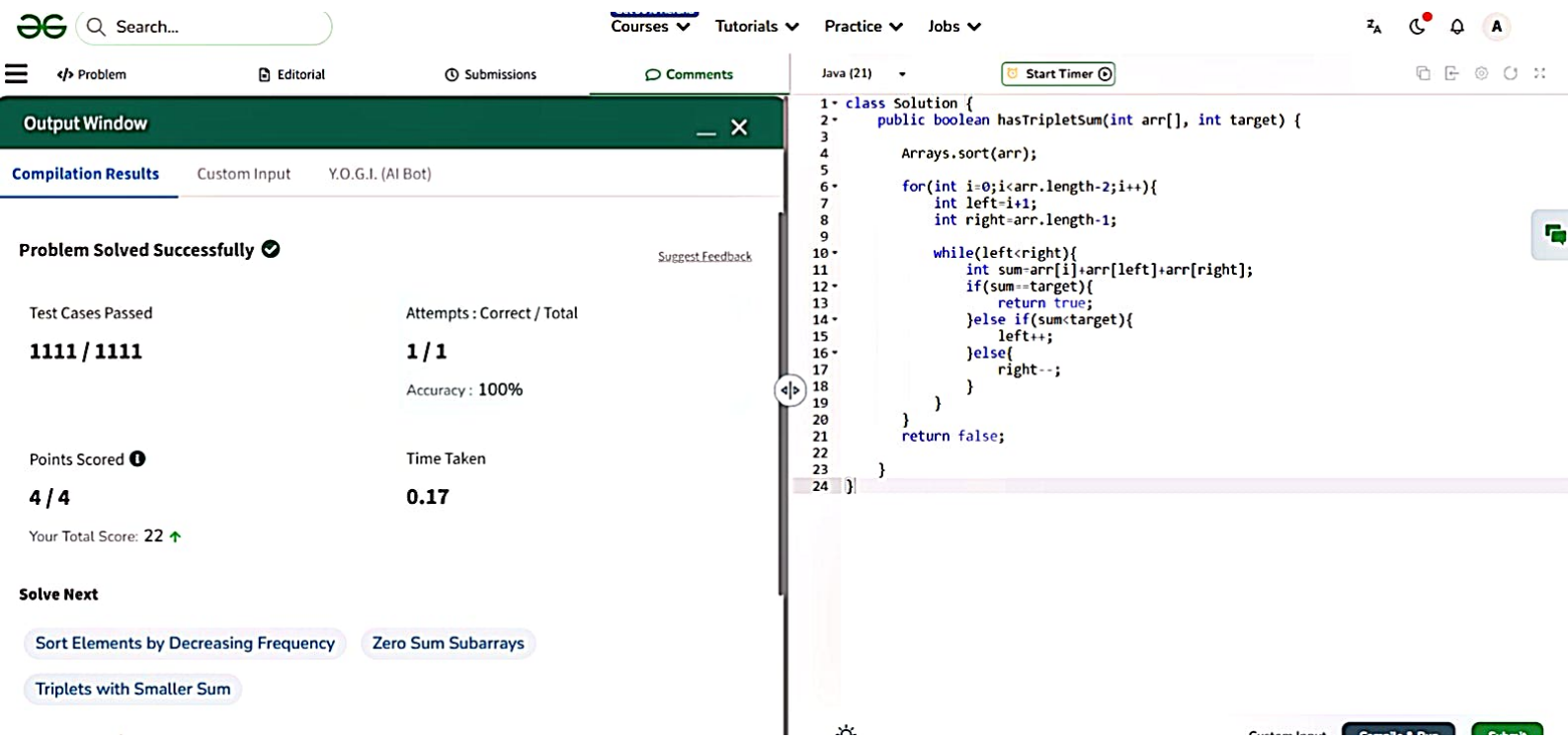

Search...
Courses
Tutorials
Practice
Jobs
Problem
Editorial
Submissions
Comments
Java (21)
Start Timer
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1111 / 1111
Attempts : Correct / Total
1 / 1
Accuracy : 100%
Points Scored
4 / 4
Time Taken
0.69
Your Total Score: 11
Solve Next
Median of 2 Sorted Arrays of Different Sizes
Nth Natural Number
Smallest Positive Integer that can not be represented as Sum
Stay Ahead With:
if (i < n && j < n) {
int temp = a[i];
a[i] = a[j];
a[j] = temp;
} else if (i < n) {
int temp = a[i];
a[i] = b[j - n];
b[j - n] = temp;
} else {
int temp = b[i - n];
b[i - n] = b[j - n];
b[j - n] = temp;
}
}
i++;
j++;
}
if (gap == 1) break;
gap = (gap + 1) / 2;
}
}
public static void main(String[] args) {
Solution ob = new Solution();
int[] a = {2, 4, 7, 10};
int[] b = {2, 3};
ob.mergeArrays(a, b);
for (int x : a) System.out.print(x + " ");
System.out.println();
for (int x : b) System.out.print(x + " ");
}
}
Custom Input
Compile & Run
Submit

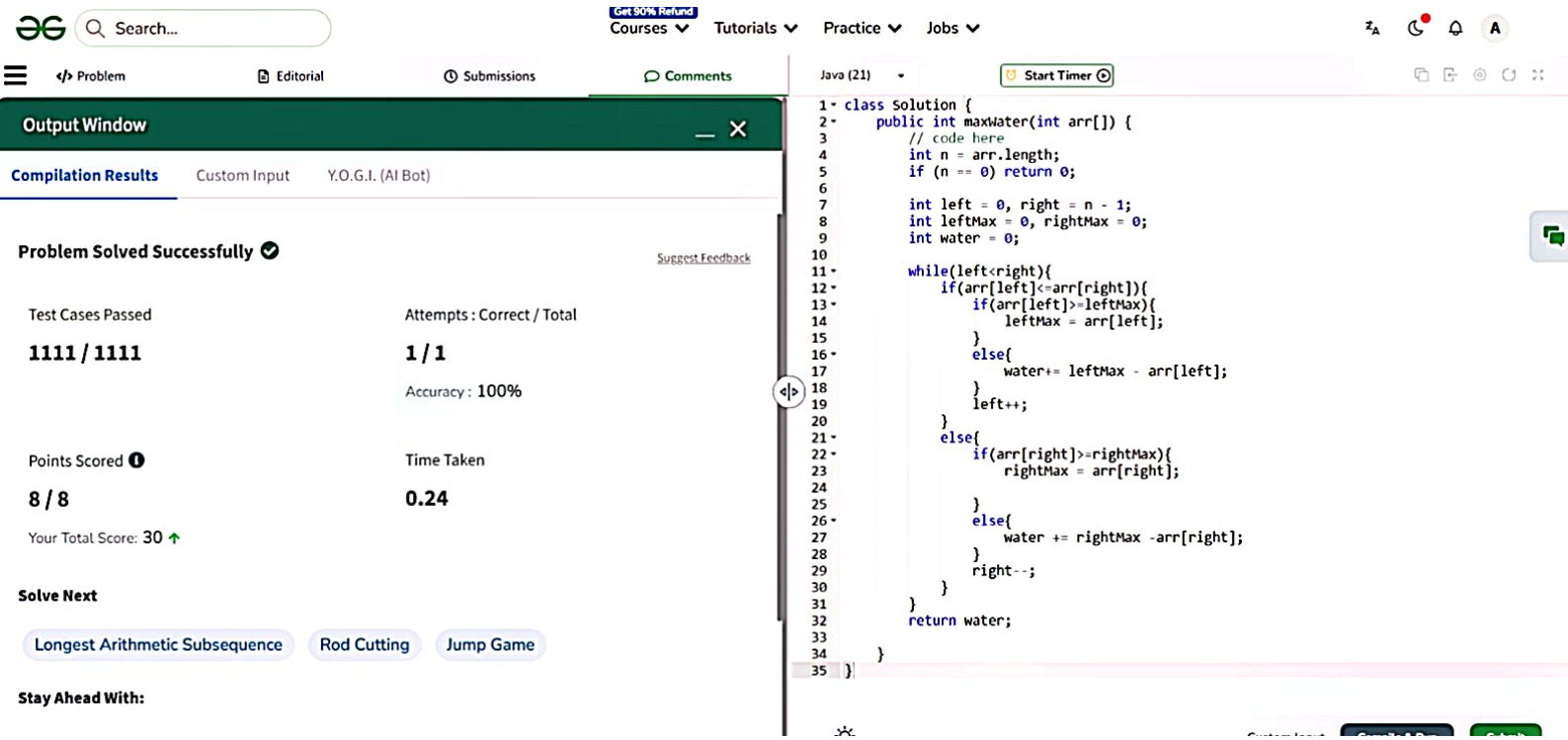

Problem List
Submit
Premium
Description | Accepted | Editorial | Solutions | Submissions
All Submissions
Accepted 172 / 172 testcases passed
VWv6VKUnQ submitted at Feb 07, 2026 23:15
Editorial
Solution
Runtime
8 ms | Beats 90.14%
Analyze Complexity
Memory
49.48 MB | Beats 11.61%
75%
50%
25%
0%
0.05% of solutions used 4 ms of runtime
2ms
4ms
6ms
8ms
10ms
12ms
14ms
Code | Java
1 import java.util.*;
2
3 class Solution {
4 public int[][] merge(int[][] intervals) {
5 if (intervals.length <= 1)
Code
Java
Auto
10 List<int[]> result = new ArrayList<>();
11 int[] current = intervals[0];
12
13 for (int i = 1; i < intervals.length; i++) {
14 if (current[1] >= intervals[i][0]) {
15 current[1] = Math.max(current[1], intervals[i][1]);
16 } else {
17 result.add(current);
18 current = intervals[i];
19 }
20 }
21 result.add(current);
22
23 return result.toArray(new int[result.size()][]);
24 }
25
26 public static void main(String[] args) {
27 Solution s = new Solution();
28
29 int[][] intervals1 = {{1,3},{2,6},{8,10},{15,18}};
30 System.out.println(Arrays.deepToString(s.merge(intervals1)));
31
32 int[][] intervals2 = {{1,4},{4,5}};
33 System.out.println(Arrays.deepToString(s.merge(intervals2)));
34
35 int[][] intervals3 = {{4,7},{1,4}};
36 System.out.println(Arrays.deepToString(s.merge(intervals3)));
37 }
Saved
Ln 39, Col 1
Testcase | Test Result

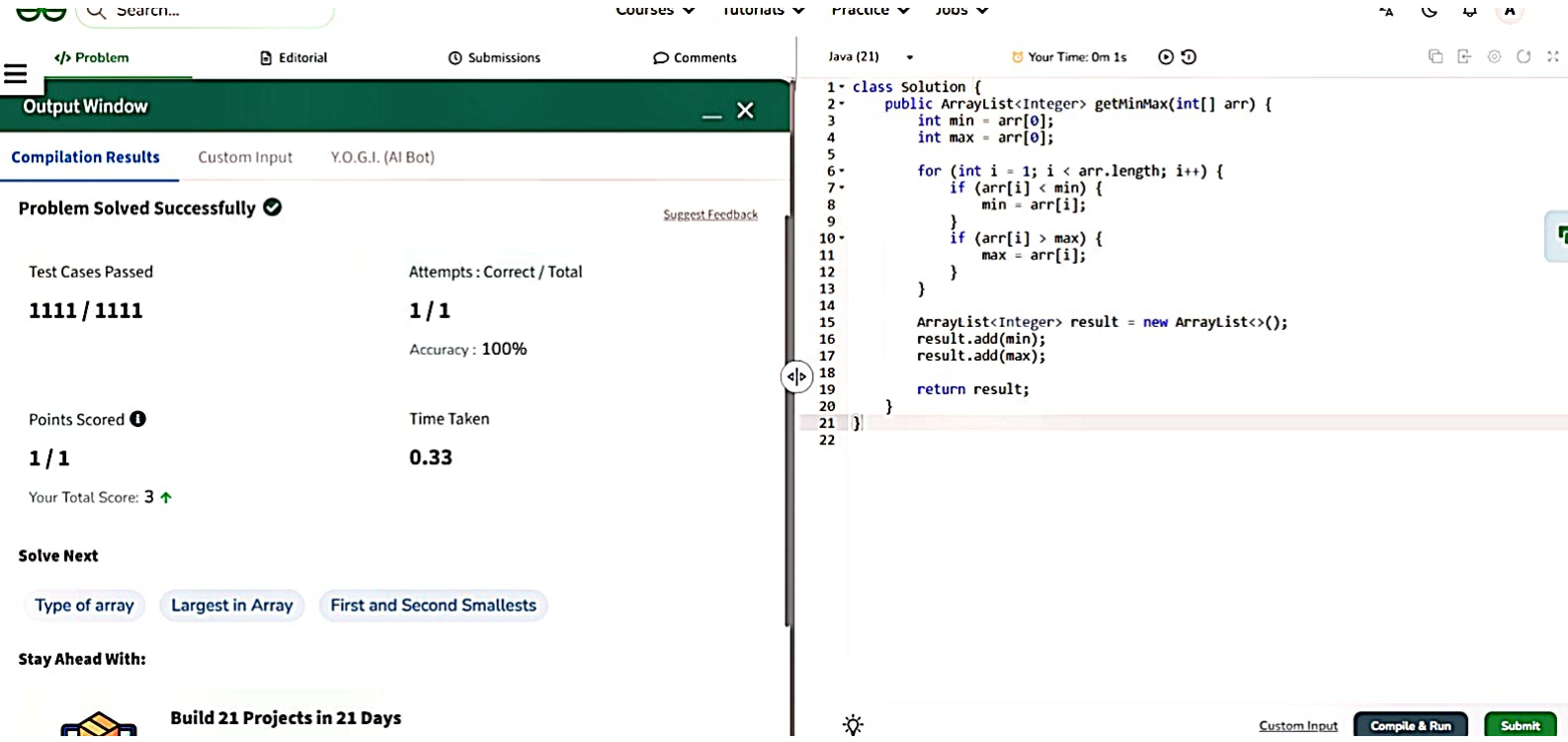

Search...
Courses
Tutorials
Practice
Jobs
Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1215 / 1215
Attempts : Correct / Total
1 / 1
Accuracy : 100%
Points Scored
2 / 2
Time Taken
3.73
Your Total Score: 13
Solve Next
Two Repeated Elements
Sorted and Rotated Minimum
Sorted Insert Position
Stay Ahead With:
Java (21)
Start Timer
// User function Template for Java
class Solution {
// Function to find common elements in three arrays.
public List<Integer> commonElements(List<Integer> arr1, List<Integer> arr2,
List<Integer> arr3) {
Map<Integer,Integer> mp = new TreeMap<>();
HashSet<Integer> h1 = new HashSet<>(arr1);
HashSet<Integer> h2 = new HashSet<>(arr2);
HashSet<Integer> h3 = new HashSet<>(arr3);
for(int i :h1){
mp.put(i,mp.getOrDefault(i,0)+1);
}
for(int i :h2){
mp.put(i,mp.getOrDefault(i,0)+1);
}
for(int i :h3){
mp.put(i,mp.getOrDefault(i,0)+1);
}
List<Integer> res= new ArrayList<>();
for(Map.Entry<Integer,Integer> entry: mp.entrySet()){
if(entry.getValue()==3){
res.add(entry.getKey());
}
}
return res;
}
}
Custom Input
Compile & Run
Submit

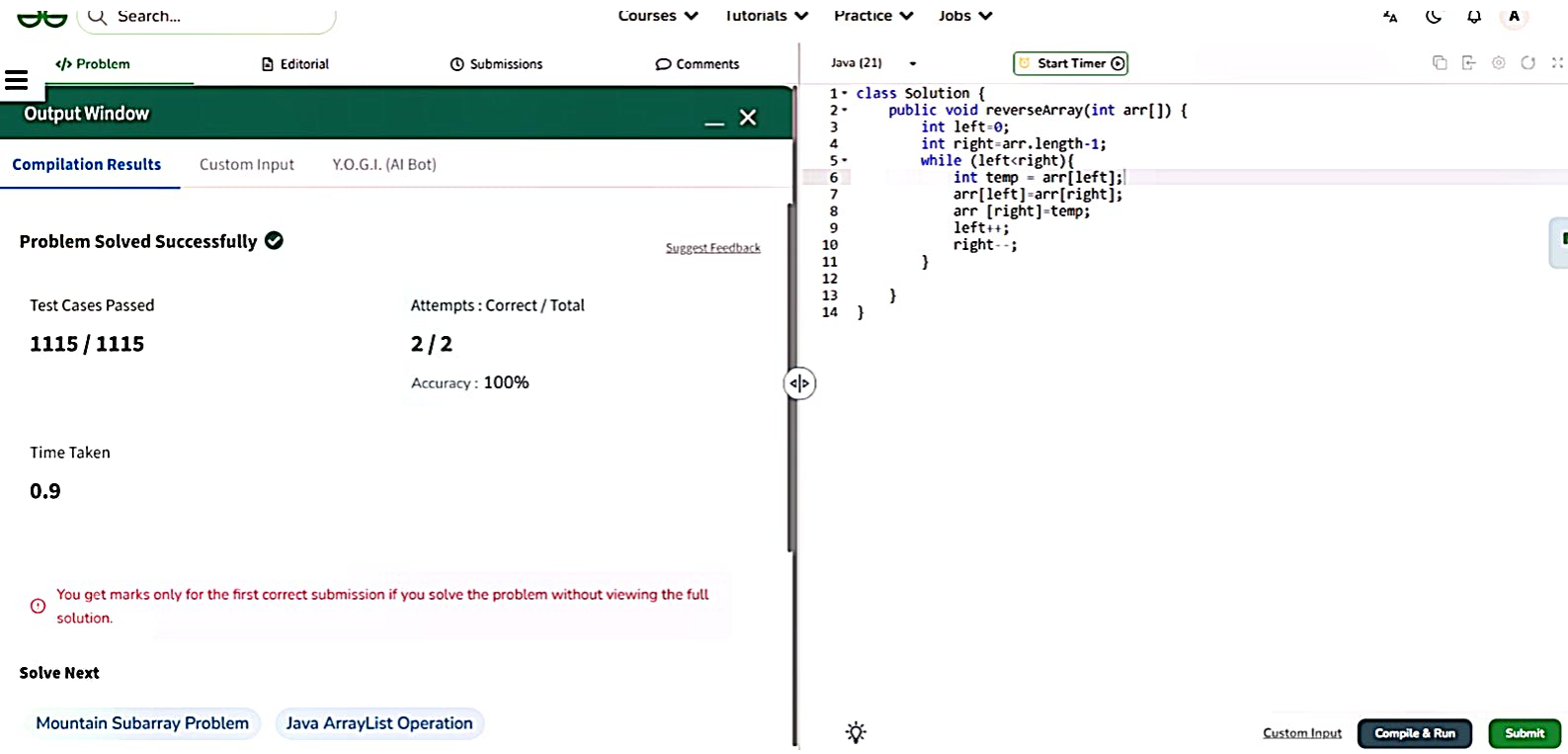

Search...
Courses
Tutorials
Practice
Jobs
Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1111 / 1111
Attempts : Correct / Total
1 / 1
Accuracy : 100%
Points Scored
4 / 4
Time Taken
0.52
Your Total Score: 17
Solve Next
Large Factorial
Number following a pattern
Rank The Permutations
Stay Ahead With:
Java (21)
Start Timer
class Solution {
public static ArrayList<Integer> factorial(int n) {
ArrayList<Integer>res=new ArrayList<>();
res.add(1);
for(int x=2;x<=n;x++){
multiply(x,res);
}
Collections.reverse(res);
return res;
}
private static void multiply(int x,ArrayList<Integer>res){
int carry=0;
for(int i=0;i<res.size();i++){
int prod=res.get(i)*x+carry;
res.set(i,prod%10);
carry=prod/10;
}
while(carry!=0){
res.add(carry%10);
carry/=10;
}
}
}
Custom Input
Compile & Run
Submit


Search...
Courses
Tutorials
Practice
Jobs
Problem
Editorial
Submissions
Comments
Java (21)
Start Timer
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1114 / 1114
Attempts : Correct / Total
1 / 1
Accuracy : 100%
Points Scored
1 / 1
Time Taken
0.56
Your Total Score: 18
Solve Next
Counting elements in two arrays
Union of 2 Sorted Arrays
Left most and right most index
class Solution {
public boolean isSubset(int a[], int b[]) {
HashMap<Integer,Integer> map = new HashMap<>();
for(int i=0;i<a.length;i++)
{
map.put(a[i],map.getOrDefault(a[i],0)+1);
}
for(int i=0;i<b.length;i++)
{
if(map.get(b[i]) !=null && map.get(b[i])>0)
map.put(b[i],map.getOrDefault(b[i],0)-1);
else
{
return false;
}
}
return true;
}
}
Custom Input
Compile & Run
Submit


Search...
Courses
Tutorials
Practice
Jobs
Problem
Editorial
Submissions
Comments
Java (21)
Start Timer
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1111 / 1111
Attempts : Correct / Total
1 / 1
Accuracy : 100%
Points Scored
4 / 4
Time Taken
0.17
Your Total Score: 22
Solve Next
Sort Elements by Decreasing Frequency
Zero Sum Subarrays
Triplets with Smaller Sum
class Solution {
public boolean hasTripletSum(int arr[], int target) {
Arrays.sort(arr);
for(int i=0;i<arr.length-2;i++){
int left=i+1;
int right=arr.length-1;
while(left<right){
int sum=arr[i]+arr[left]+arr[right];
if(sum==target){
return true;
}else if(sum<target){
left++;
}else{
right--;
}
}
}
return false;
}
}


Search...
Get 90% Refund
Courses
Tutorials
Practice
Jobs
Problem
Editorial
Submissions
Comments
Java (21)
Start Timer
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1111 / 1111
Attempts : Correct / Total
1 / 1
Accuracy : 100%
Points Scored
8 / 8
Time Taken
0.24
Your Total Score: 30
Solve Next
Longest Arithmetic Subsequence
Rod Cutting
Jump Game
Stay Ahead With:
class Solution {
public int maxWater(int arr[]) {
// code here
int n = arr.length;
if (n == 0) return 0;
int left = 0, right = n - 1;
int leftMax = 0, rightMax = 0;
int water = 0;
while(left<right){
if(arr[left]<=arr[right]){
if(arr[left]>=leftMax){
leftMax = arr[left];
}
else{
water+= leftMax - arr[left];
}
left++;
}
else{
if(arr[right]>=rightMax){
rightMax = arr[right];
}
else{
water += rightMax -arr[right];
}
right--;
}
}
return water;
}
}


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
1111 / 1111
Attempts : Correct / Total
1 / 1
Accuracy : 100%
Points Scored
1 / 1
Time Taken
0.33
Your Total Score: 3
Solve Next
Type of array
Largest in Array
First and Second Smallests
Stay Ahead With:
Build 21 Projects in 21 Days
Java (21)
Your Time: 0m 1s
class Solution {
public ArrayList<Integer> getMinMax(int[] arr) {
int min = arr[0];
int max = arr[0];
for (int i = 1; i < arr.length; i++) {
if (arr[i] < min) {
min = arr[i];
}
if (arr[i] > max) {
max = arr[i];
}
}
ArrayList<Integer> result = new ArrayList<>();
result.add(min);
result.add(max);
return result;
}
}
Custom Input
Compile & Run
Submit


Search...
Courses
Tutorials
Practice
Jobs
Problem
Editorial
Submissions
Comments
Java (21)
Start Timer
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
2 / 2
Accuracy : 100%
Time Taken
0.9
You get marks only for the first correct submission if you solve the problem without viewing the full solution.
Solve Next
Mountain Subarray Problem
Java ArrayList Operation
class Solution {
public void reverseArray(int arr[]) {
int left=0;
int right=arr.length-1;
while (left<right){
int temp = arr[left];
arr[left]=arr[right];
arr [right]=temp;
left++;
right--;
}
}
}
Custom Input
Compile & Run
Submit