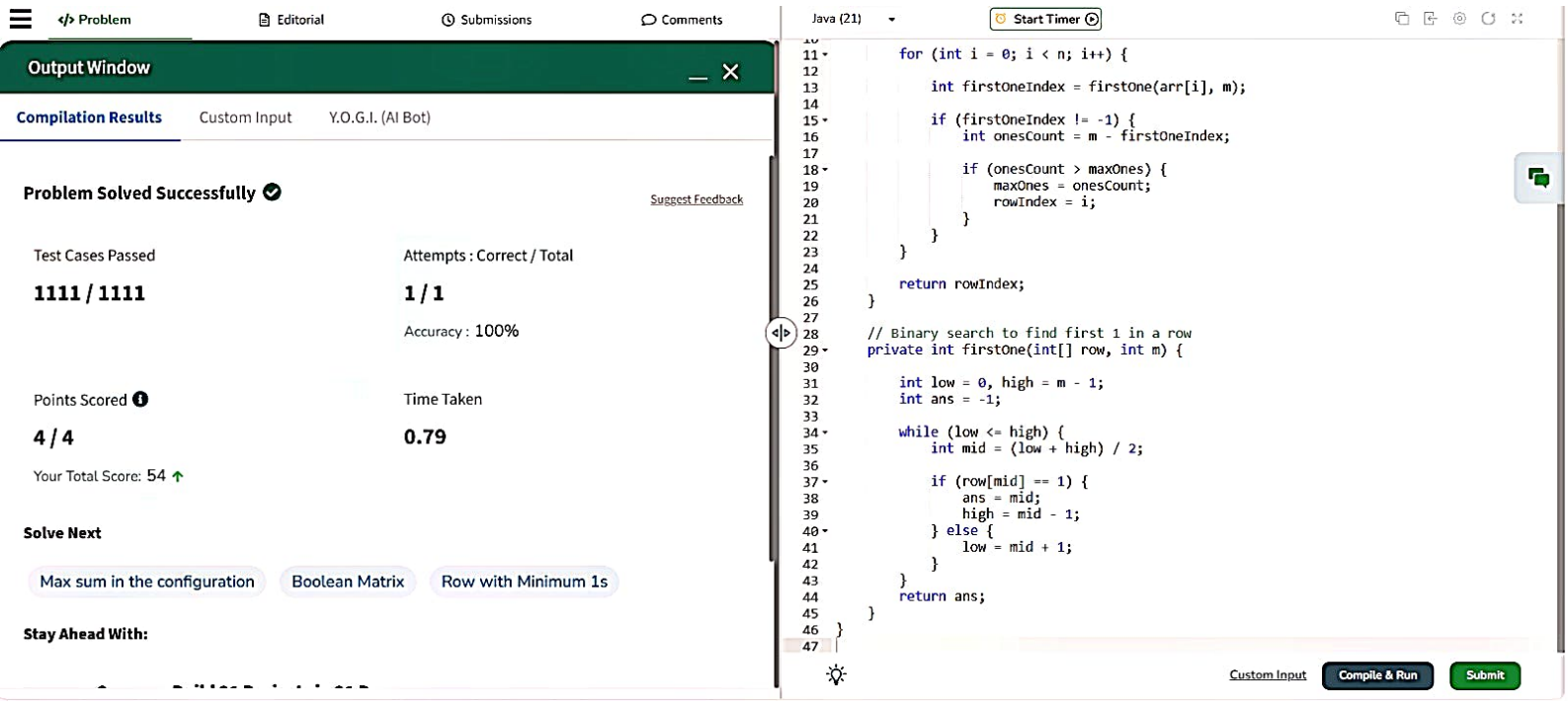
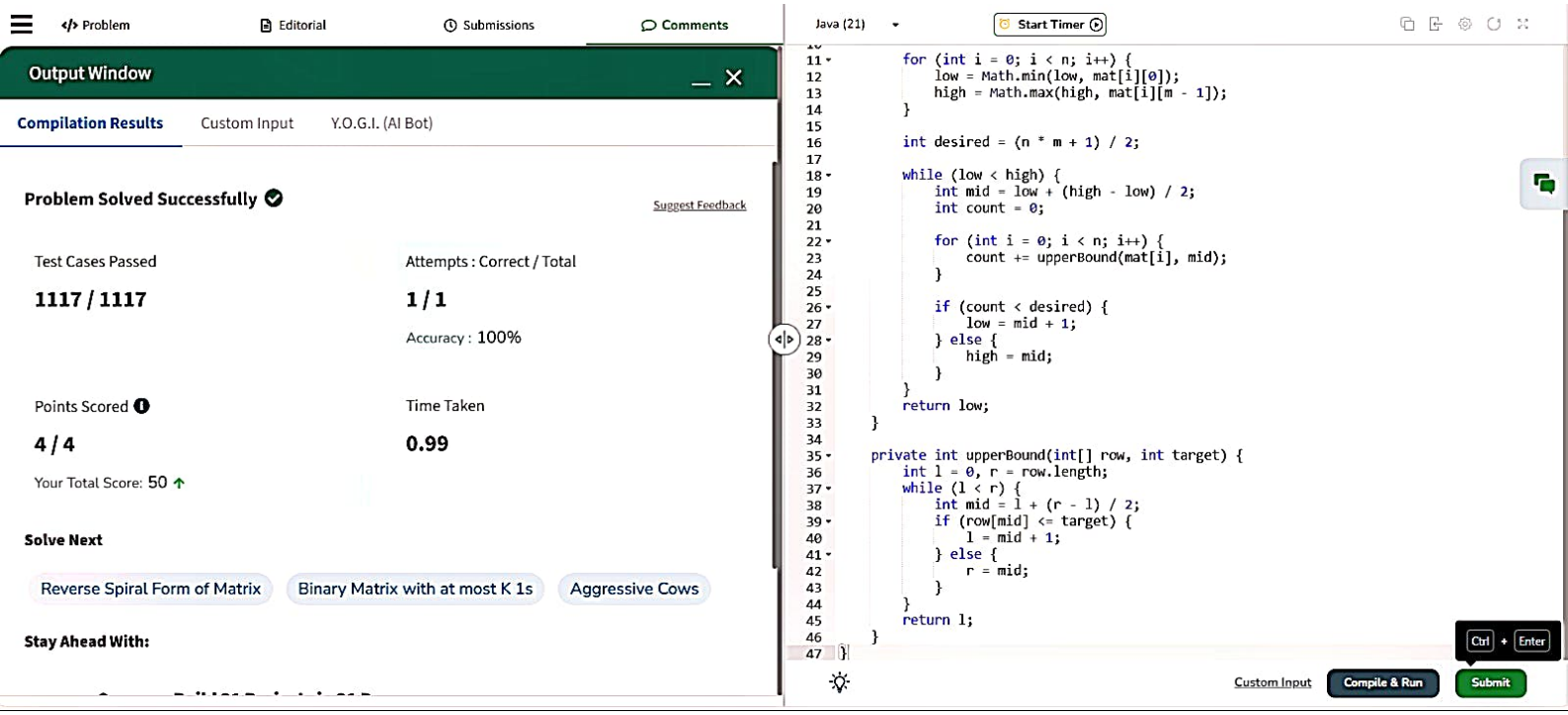

Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1111 / 1111
Attempts : Correct / Total
1 / 1
Accuracy : 100%
Points Scored
4 / 4
Time Taken
0.79
Your Total Score: 54
Solve Next
Max sum in the configuration
Boolean Matrix
Row with Minimum 1s
Stay Ahead With:
Java (21)
Start Timer
for (int i = 0; i < n; i++) {
int firstOneIndex = firstOne(arr[i], m);
if (firstOneIndex != -1) {
int onesCount = m - firstOneIndex;
if (onesCount > maxOnes) {
maxOnes = onesCount;
rowIndex = i;
}
}
}
return rowIndex;
}
// Binary search to find first 1 in a row
private int firstOne(int[] row, int m) {
int low = 0, high = m - 1;
int ans = -1;
while (low <= high) {
int mid = (low + high) / 2;
if (row[mid] == 1) {
ans = mid;
high = mid - 1;
} else {
low = mid + 1;
}
}
return ans;
}
}
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
1117 / 1117
Attempts : Correct / Total
1 / 1
Accuracy : 100%
Points Scored
4 / 4
Time Taken
0.99
Your Total Score: 50
Solve Next
Reverse Spiral Form of Matrix
Binary Matrix with at most K 1s
Aggressive Cows
Stay Ahead With:
Java (21)
Start Timer
for (int i = 0; i < n; i++) {
low = Math.min(low, mat[i][0]);
high = Math.max(high, mat[i][m - 1]);
}
int desired = (n * m + 1) / 2;
while (low < high) {
int mid = low + (high - low) / 2;
int count = 0;
for (int i = 0; i < n; i++) {
count += upperBound(mat[i], mid);
}
if (count < desired) {
low = mid + 1;
} else {
high = mid;
}
}
return low;
}
private int upperBound(int[] row, int target) {
int l = 0, r = row.length;
while (l < r) {
int mid = l + (r - l) / 2;
if (row[mid] <= target) {
l = mid + 1;
} else {
r = mid;
}
}
return l;
}
}
Ctrl + Enter
Custom Input
Compile & Run
Submit

Description | Accepted × | Editorial | Solutions | Submissions

All Submissions

**Accepted** 133 / 133 testcases passed

amit_89333 submitted at Feb 08, 2026 00:18

Editorial | Solution

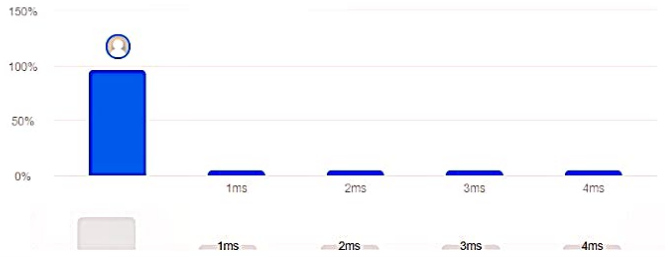
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

44.08 MB Beats 39.09%




Code | Java

```java
class Solution {
    public boolean searchMatrix(int[][] matrix, int target) {
        int m = matrix.length;
        int n = matrix[0].length;

```

Code

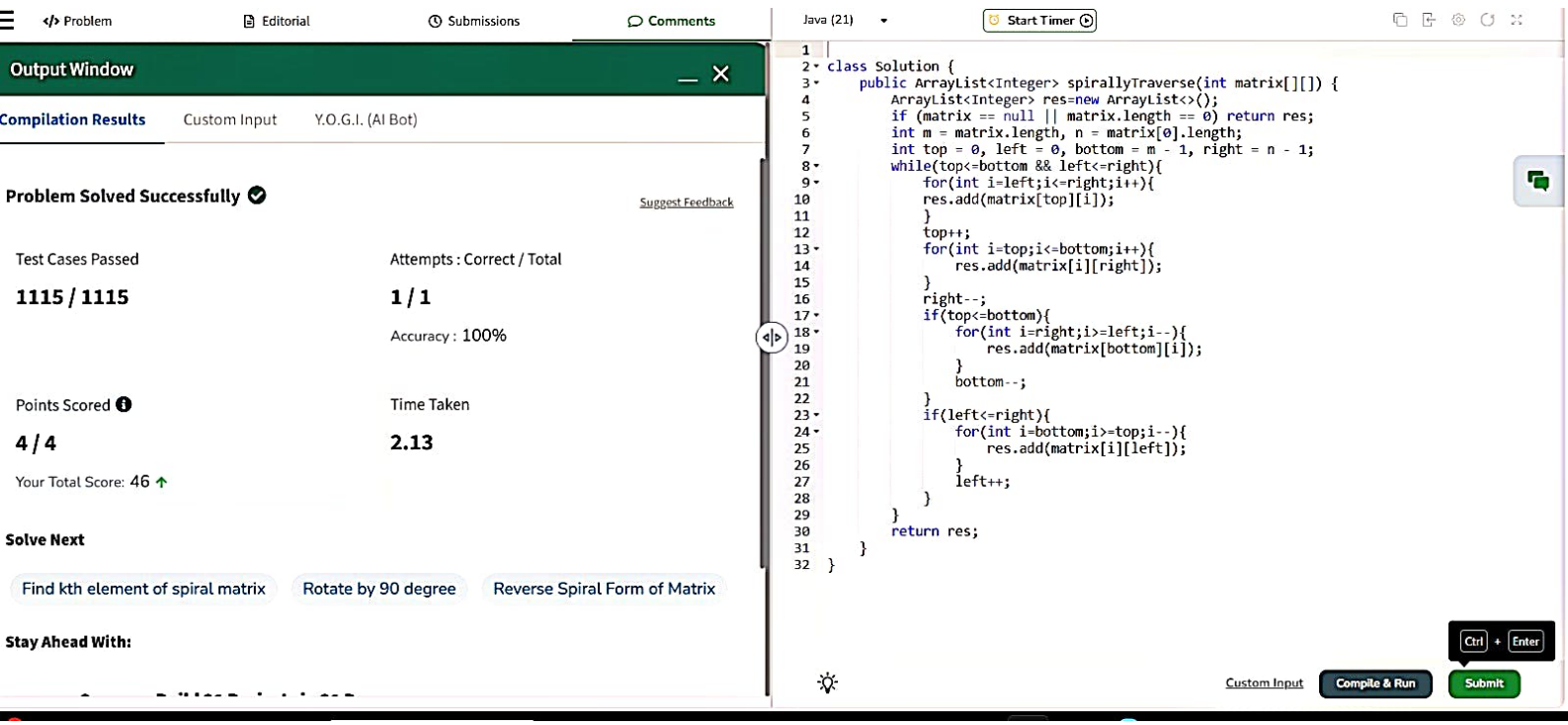
Java | Auto

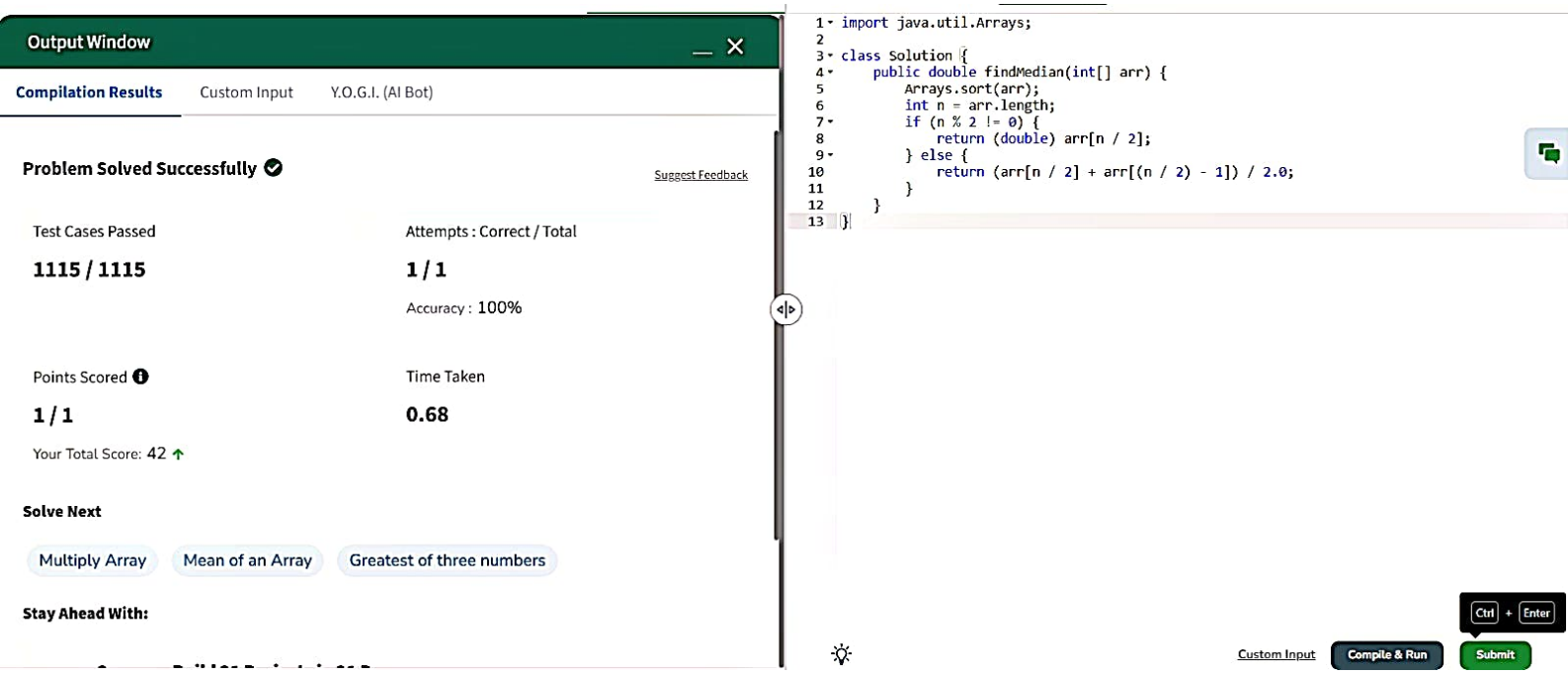
```java
class Solution {
    public boolean searchMatrix(int[][] matrix, int target) {
        int m = matrix.length;
        int n = matrix[0].length;

        int left = 0;
        int right = m * n - 1;

        while (left <= right) {
            int mid = left + (right - left) / 2;
            int row = mid / n;
            int col = mid % n;

            if (matrix[row][col] == target) {
                return true;
            }
            else if (matrix[row][col] < target) {
                left = mid + 1;
            }
            else {
                right = mid - 1;
            }
        }
        return false;
    }
}
```

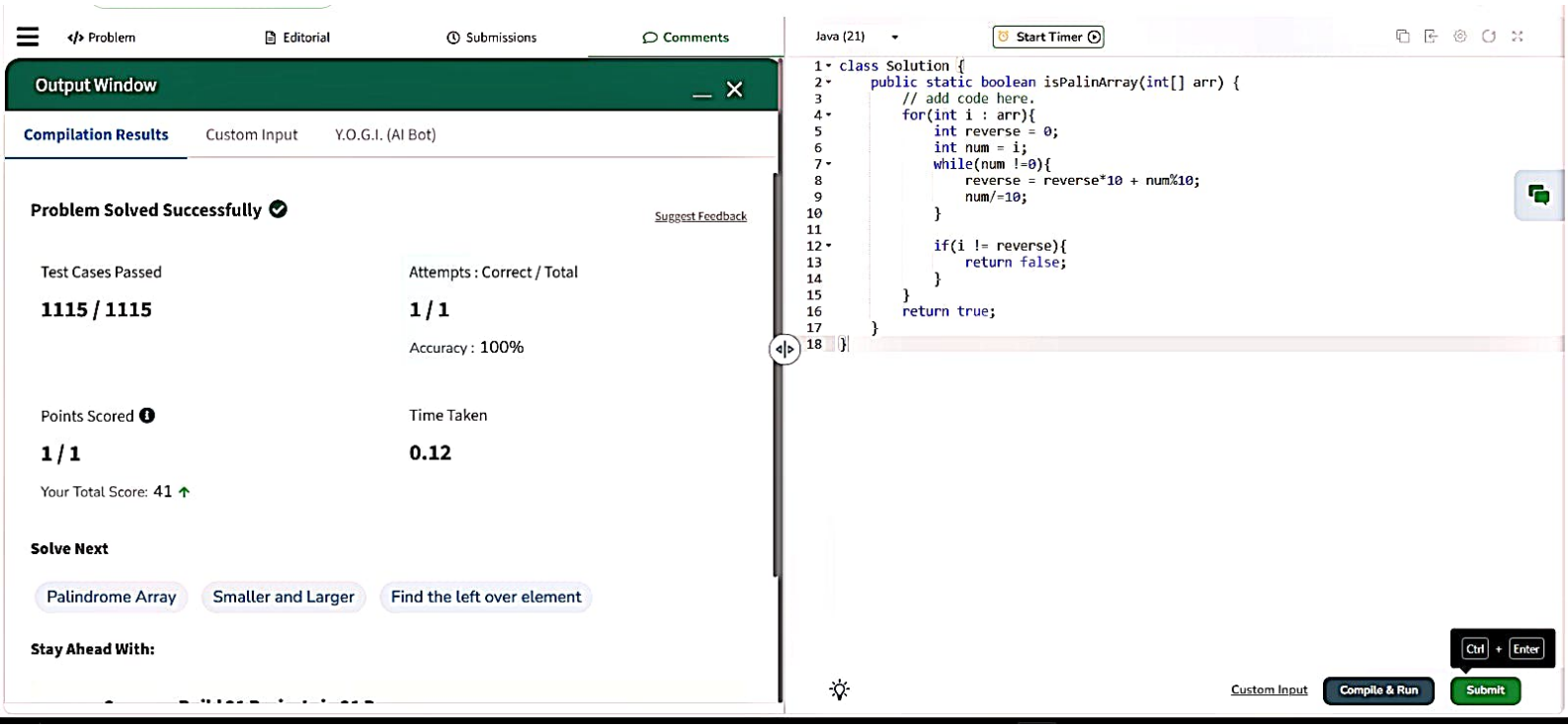
Saved

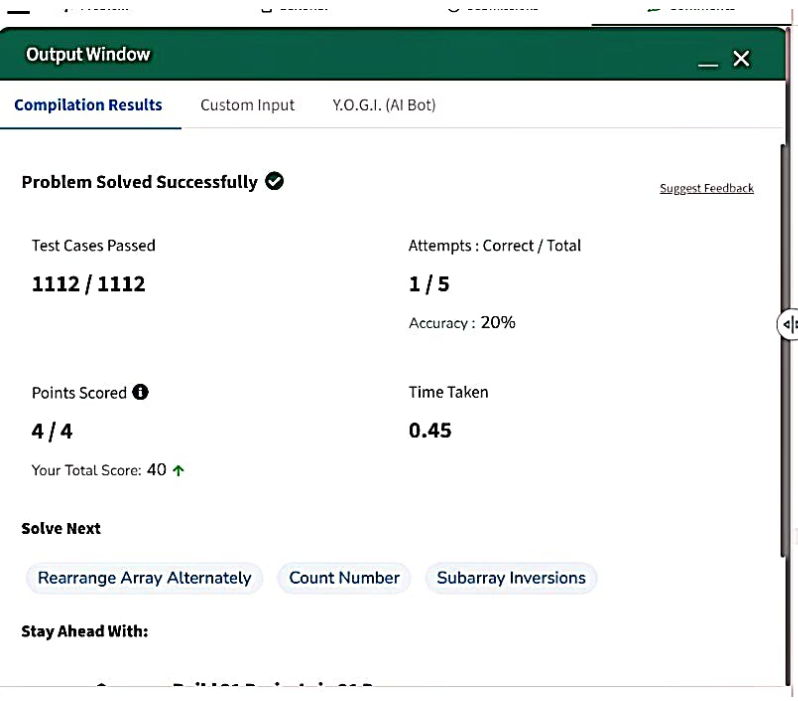
Ln 26, Col 2

Testcase | Test Result

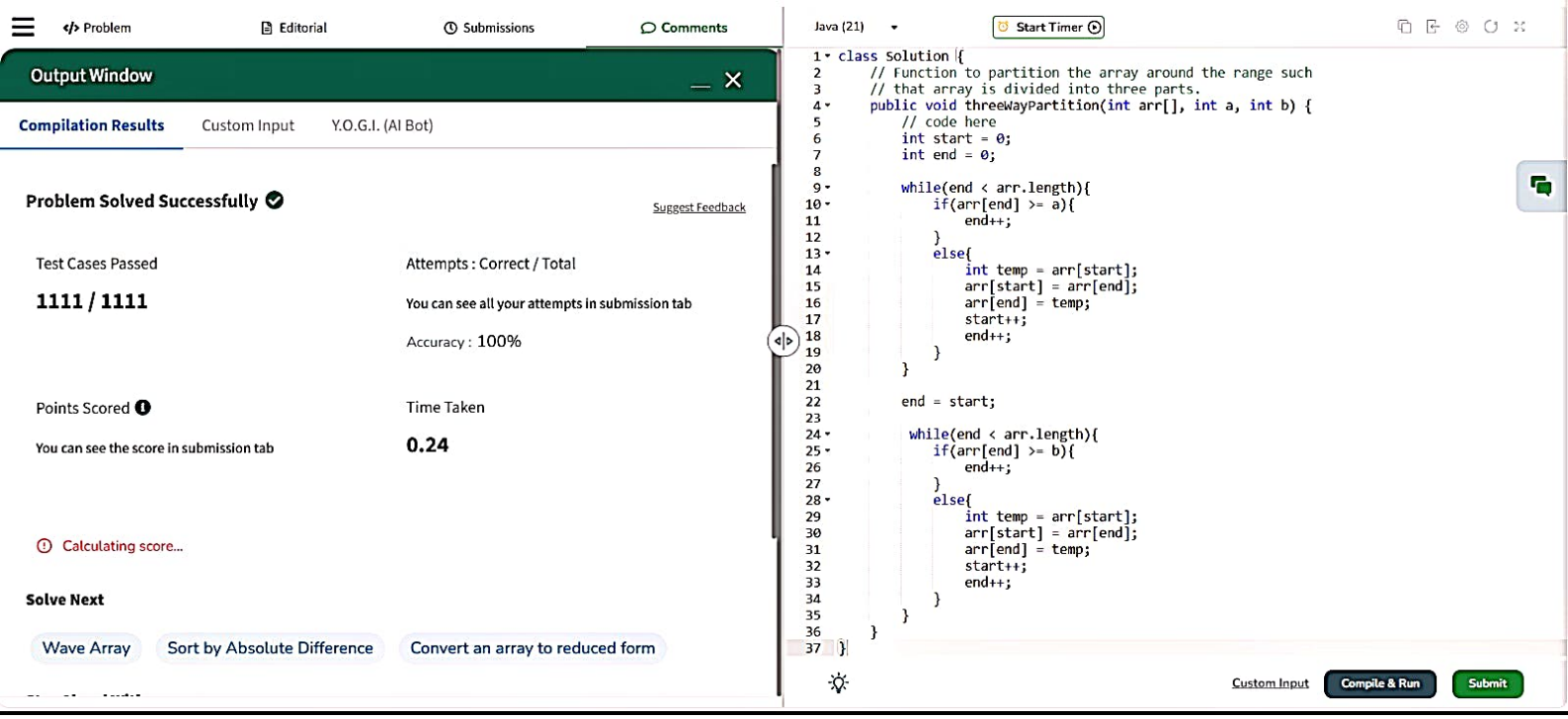

Problem
Editorial
Submissions
Comments
Java (21)
Start Timer
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1115 / 1115
Attempts : Correct / Total
1 / 1
Accuracy : 100%
Points Scored
4 / 4
Time Taken
2.13
Your Total Score: 46
Solve Next
Find kth element of spiral matrix
Rotate by 90 degree
Reverse Spiral Form of Matrix
Stay Ahead With:
class Solution {
public ArrayList<Integer> spirallyTraverse(int matrix[][]) {
ArrayList<Integer> res=new ArrayList<>();
if (matrix == null || matrix.length == 0) return res;
int m = matrix.length, n = matrix[0].length;
int top = 0, left = 0, bottom = m - 1, right = n - 1;
while(top<=bottom && left<=right){
for(int i=left;i<=right;i++){
res.add(matrix[top][i]);
}
top++;
for(int i=top;i<=bottom;i++){
res.add(matrix[i][right]);
}
right--;
if(top<=bottom){
for(int i=right;i>=left;i--){
res.add(matrix[bottom][i]);
}
bottom--;
}
if(left<=right){
for(int i=bottom;i>=top;i--){
res.add(matrix[i][left]);
}
left++;
}
}
return res;
}
}
Ctrl + Enter
Custom Input
Compile & Run
Submit

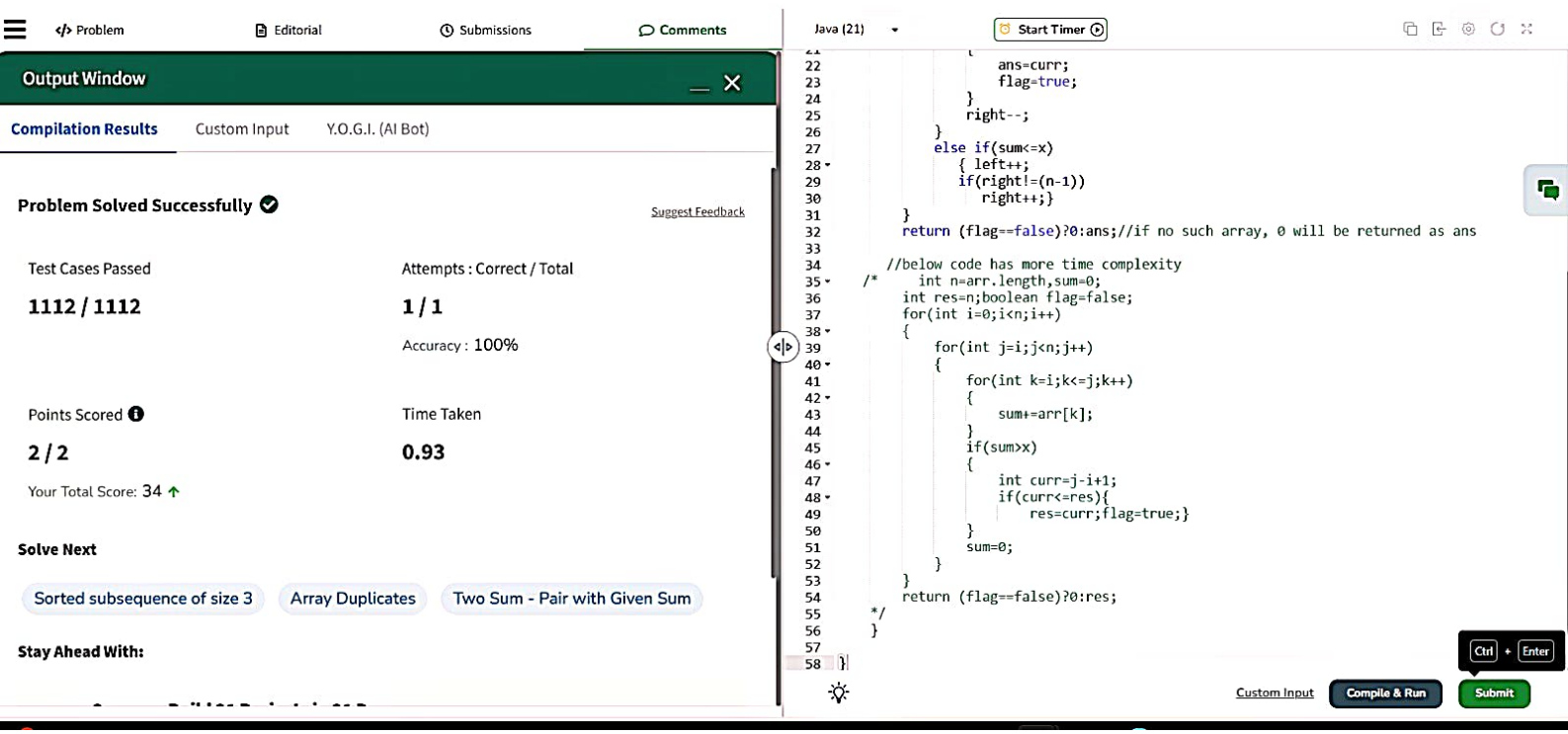

Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1115 / 1115
Attempts : Correct / Total
1 / 1
Accuracy : 100%
Points Scored
1 / 1
Time Taken
0.68
Your Total Score: 42
Solve Next
Multiply Array
Mean of an Array
Greatest of three numbers
Stay Ahead With:
1 import java.util.Arrays;
2
3 class Solution {
4 public double findMedian(int[] arr) {
5 Arrays.sort(arr);
6 int n = arr.length;
7 if (n % 2 != 0) {
8 return (double) arr[n / 2];
9 } else {
10 return (arr[n / 2] + arr[(n / 2) - 1]) / 2.0;
11 }
12 }
13 }
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
1 / 1
Time Taken
0.12
Your Total Score: 41
Solve Next
Palindrome Array
Smaller and Larger
Find the left over element
Stay Ahead With:
Java (21)
Start Timer
class Solution {
public static boolean isPalinArray(int[] arr) {
// add code here.
for(int i : arr){
int reverse = 0;
int num = i;
while(num !=0){
reverse = reverse*10 + num%10;
num/=10;
}
if(i != reverse){
return false;
}
}
return true;
}
}
Ctrl + Enter
Custom Input
Compile & Run
Submit




```java
class Solution {
    public static int minSwap(int[] arr, int k) {
        int n = arr.length;

        // Count elements <= k (window size)
        int good = 0;
        for (int x : arr) {
            if (x <= k) good++;
        }

        if (good == 0 || good == n) return 0;

        // Count bad elements in first window
        int bad = 0;
        for (int i = 0; i < good; i++) {
            if (arr[i] > k) bad++;
        }

        int minSwaps = bad;

        // Slide the window
        for (int i = 0, j = good; j < n; i++, j++) {
            if (arr[i] > k) bad--;
            if (arr[j] > k) bad++;

            minSwaps = Math.min(minSwaps, bad);
        }

        return minSwaps;
    }
}
```


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
You can see all your attempts in submission tab
Accuracy : 100%
Points Scored
You can see the score in submission tab
Time Taken
0.24
Calculating score...
Solve Next
Wave Array
Sort by Absolute Difference
Convert an array to reduced form
Java (21)
Start Timer
class Solution {
// Function to partition the array around the range such
// that array is divided into three parts.
public void threeWayPartition(int arr[], int a, int b) {
// code here
int start = 0;
int end = 0;
while(end < arr.length){
if(arr[end] >= a){
end++;
}
else{
int temp = arr[start];
arr[start] = arr[end];
arr[end] = temp;
start++;
end++;
}
}
end = start;
while(end < arr.length){
if(arr[end] >= b){
end++;
}
else{
int temp = arr[start];
arr[start] = arr[end];
arr[end] = temp;
start++;
end++;
}
}
}
}
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
1112 / 1112
Attempts : Correct / Total
1 / 1
Accuracy : 100%
Points Scored
2 / 2
Time Taken
0.93
Your Total Score: 34
Solve Next
Sorted subsequence of size 3
Array Duplicates
Two Sum - Pair with Given Sum
Stay Ahead With:
Java (21)
Start Timer
ans=curr;
flag=true;
right--;
else if(sum<=x)
{ left++;
if(right!=(n-1))
right++;}
return (flag==false)?0:ans;//if no such array, 0 will be returned as ans
//below code has more time complexity
/* int n=arr.length,sum=0;
int res=n;boolean flag=false;
for(int i=0;i<n;i++)
{
for(int j=i;j<n;j++)
{
for(int k=i;k<=j;k++)
{
sum+=arr[k];
}
if(sum>x)
{
int curr=j-i+1;
if(curr<=res){
res=curr;flag=true;}
}
sum=0;
}
}
return (flag==false)?0:res;
*/
}
Ctrl + Enter
Custom Input
Compile & Run
Submit

Problem | Editorial | Submissions | Comments

Java (21) | Start Timer

```java
class Solution {
    public int findMinDiff(ArrayList<Integer> arr, int m) {
        // your code here
        Collections.sort(arr);

        int i = 0; int j = m-1;
        int n = arr.size();
        int ans = Integer.MAX_VALUE;

        while(j < n){
            ans = Math.min(ans, arr.get(j)-arr.get(i));
            i++; j++;
        }
        return ans;
    }
}
```

**Output Window**

Compilation Results | Custom Input | Y.O.G.I. (AI Bot)

**Problem Solved Successfully**

Suggest Feedback

Test Cases Passed

**1112 / 1112**

Attempts : Correct / Total

**1 / 1**

Accuracy : 100%

Points Scored

**2 / 2**

Time Taken

**0.82**

Your Total Score: 32

**Solve Next**

Bubble Sort | Floor in a Sorted Array | Closest Triplet

**Stay Ahead With:**