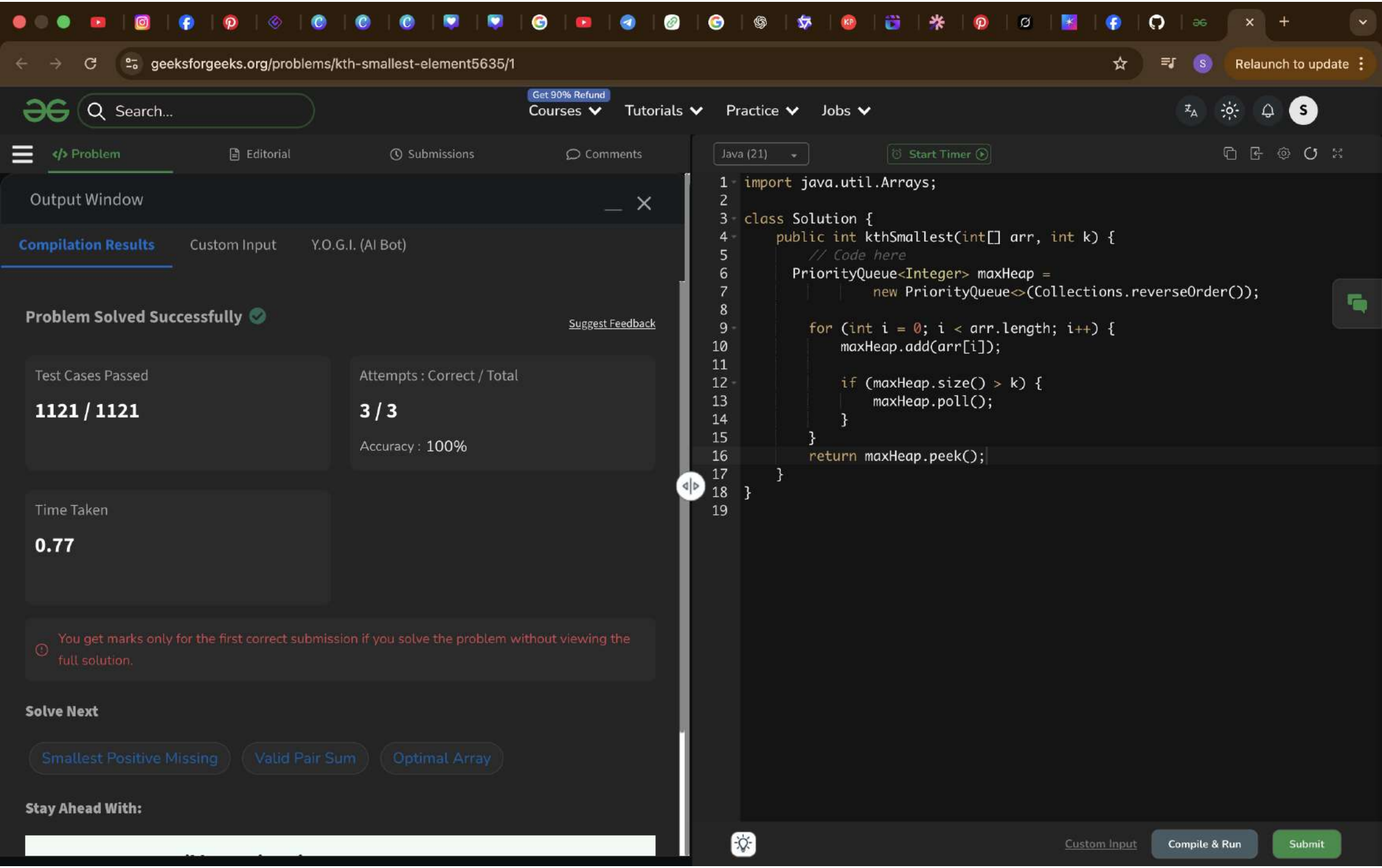

geeksforgeeks.org/problems/kth-smallest-element5635/1
Relaunch to update
Search...
Get 90% Refund
Courses
Tutorials
Practice
Jobs
Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1121 / 1121
Attempts : Correct / Total
3 / 3
Accuracy : 100%
Time Taken
0.77
You get marks only for the first correct submission if you solve the problem without viewing the full solution.
Solve Next
Smallest Positive Missing
Valid Pair Sum
Optimal Array
Stay Ahead With:
Java (21)
Start Timer
import java.util.Arrays;
class Solution {
public int kthSmallest(int[] arr, int k) {
// Code here
PriorityQueue<Integer> maxHeap =
new PriorityQueue<>(Collections.reverseOrder());
for (int i = 0; i < arr.length; i++) {
maxHeap.add(arr[i]);
if (maxHeap.size() > k) {
maxHeap.poll();
}
}
return maxHeap.peek();
}
}
Custom Input
Compile & Run
Submit

geeksforgeeks.org/problems/minimize-the-heights3351/1

Relaunch to update

Search...

Get 90% Refund

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 | Attempts : Correct / Total |
|---|---|
| **1115 / 1115** | **1 / 1** |
| | Accuracy : 100% |

| Points Scored | Time Taken |
|---|---|
| **0 / 4** | **0.7** |
| Your Total Score: 16 | |

**Solve Next**

A difference of values and indexes | Max Diff Elements and Indexes | Minimize the Heights I

**Stay Ahead With:**

**Build 21 Projects in 21 Days**

Java (21) | Start Timer

```java
class Solution {
    public int getMinDiff(int[] arr, int k) {
        // code here
        int n = arr.length;

        Arrays.sort(arr);

        int ans = arr[n - 1] - arr[0];

        int smallest = arr[0] + k;
        int largest = arr[n - 1] - k;

        for (int i = 1; i < n; i++) {
            int minHeight = Math.min(smallest, arr[i] - k);
            int maxHeight = Math.max(largest, arr[i - 1] + k);

            if (minHeight < 0)
                continue;

            ans = Math.min(ans, maxHeight - minHeight);
        }
        return ans;
    }
}
```

Custom Input | Compile & Run | Submit

geeksforgeeks.org/problems/minimum-number-of-jumps-1587115620/1

Relaunch to update

Search...

Get 90% Refund

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

**2 / 2**

Accuracy : 100%

Time Taken

**0.61**

You get marks only for the first correct submission if you solve the problem without viewing the full solution.

**Solve Next**

Maximum Index | Jump Game | Wine Buying and Selling

**Stay Ahead With:**

Java (21)

Start Timer

```java
class Solution {
    public int minJumps(int[] arr) {
        // code here
          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;
    }
}
```

Custom Input | Compile & Run | Submit

leetcode.com/problems/find-the-duplicate-number/submissions/1909310676/

Relaunch to update

Problem List

Submit

Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

**Accepted** 59 / 59 testcases passed

Satyam Kumar submitted at Feb 05, 2026 21:42

Editorial | Solution

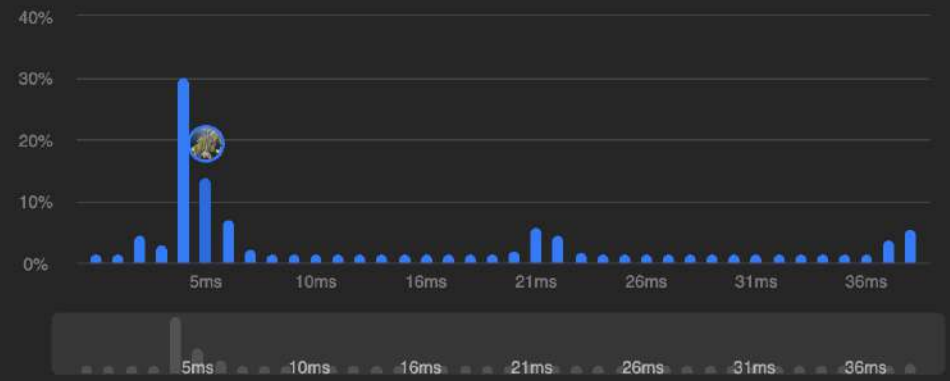
Runtime

5 ms | Beats 60.87%

Analyze Complexity

Memory

83.02 MB | Beats 42.17%




Code | Java

```java
class Solution {
    public int findDuplicate(int[] nums) {
        int slow = nums[0];
        int fast = nums[0];

        // Step 1: Detect cycle
        do {
            slow = nums[slow];
```

View more

Code

Java | Auto

```java
class Solution {
    public int findDuplicate(int[] nums) {
        int slow = nums[0];
        int fast = nums[0];

        // Step 1: Detect cycle
        do {
            slow = nums[slow];
            fast = nums[nums[fast]];
        } while (slow != fast);

        // Step 2: Find cycle entry point
```

Saved

Ln 19, Col 21

Testcase | Test Result

**Accepted** Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

nums =

[1,3,4,2,2]

Output

2

Expected

2

geeksforgeeks.org/problems/merge-two-sorted-arrays-1587115620/1

Relaunch to update

Search...

Get 90% Refund

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**

Your Total Score: 20

Time Taken

**0.59**

**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:**

**Build 21 Projects in 21 Days**

Java (21) | Start Timer

```java
class Solution {
    public void mergeArrays(int a[], int b[]) {
        // code here
         int n = a.length;
        int m = b.length;

        int gap = n + m;

        while (gap > 0) {
            gap = nextGap(gap);

            int i = 0;
            int j = gap;

            while (j < n + m) {
                // both elements in array a
                if (i < n && j < n) {
                    if (a[i] > a[j]) {
                        int temp = a[i];
                        a[i] = a[j];
                        a[j] = temp;
                    }
                }
                // i in a, j in b
                else if (i < n && j >= n) {
                    if (a[i] > b[j - n]) {
                        int temp = a[i];
                        a[i] = b[j - n];
                        b[j - n] = temp;
                    }
                }
                // both elements in array b
                else {
                    if (b[i - n] > b[j - n]) {
                        int temp = b[i - n];
```

Custom Input | Compile & Run | Submit

leetcode.com/problems/merge-intervals/submissions/1909316771/

Relaunch to update

Problem List

Submit

Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

**Accepted** 172 / 172 testcases passed

Satyam Kumar submitted at Feb 05, 2026 21:47

Editorial | Solution

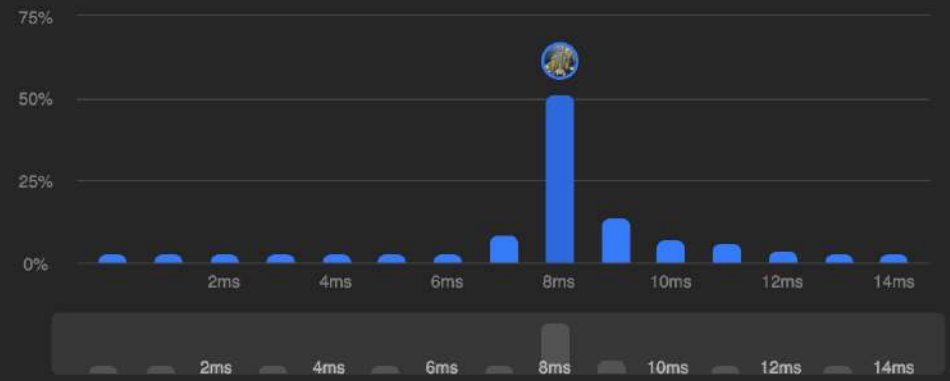
Runtime

8 ms | Beats 90.13%

Analyze Complexity

Memory

49.12 MB | Beats 52.31%




Code | Java

```java
class Solution {
    public int[][] merge(int[][] intervals) {
        if (intervals.length <= 1)
            return intervals;

        // Step 1: Sort by start time
        Arrays.sort(intervals, (a, b) -> a[0] - b[0]);

```

View more

Code

Java | Auto

```java
class Solution {
    public int[][] merge(int[][] intervals) {
        if (intervals.length <= 1)
            return intervals;

        // Step 1: Sort by start time
        Arrays.sort(intervals, (a, b) -> a[0] - b[0]);

        List<int[]> result = new ArrayList<>();

        int[] current = intervals[0];
```

Saved

Ln 26, Col 57

Testcase | Test Result

**Accepted** Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

intervals =

`[[1,3],[2,6],[8,10],[15,18]]`

Output

`[[1,6],[8,10],[15,18]]`

Expected

`[[1,6],[8,10],[15,18]]`

geeksforgeeks.org/problems/common-elements1132/1

Relaunch to update

Search...

Get 90% Refund

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 | Attempts : Correct / Total |
|---|---|
| **1215 / 1215** | **1 / 1** |
| | Accuracy : 100% |

| Points Scored | Time Taken |
|---|---|
| **2 / 2** | **3.76** |
| Your Total Score: 22 ↑ | |

**Solve Next**

Two Repeated Elements | Sorted and Rotated Minimum | Sorted Insert Position

**Stay Ahead With:**

**Build 21 Projects in 21 Days**

Java (21) | Start Timer

```java
// 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) {
        // Code Here
        List<Integer> result = new ArrayList<>();

        int i = 0, j = 0, k = 0;

        while (i < arr1.size() && j < arr2.size() && k < arr3.size()) {

            int a = arr1.get(i);
            int b = arr2.get(j);
            int c = arr3.get(k);

            // If all three are equal
            if (a == b && b == c) {

                // Avoid duplicates in result
                if (result.size() == 0 || result.get(result.size() - 1) != a) {
                    result.add(a);
                }

                i++;
                j++;
                k++;
            }
            // Move the pointer with the smallest value
            else if (a < b) {
                i++;
            }
            else if (b < c) {
                j++;
```

Custom Input | Compile & Run | Submit

geeksforgeeks.org/problems/factorials-of-large-numbers2508/1

Relaunch to update

Search...

Get 90% Refund

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 | Attempts : Correct / Total |
|---|---|
| **1111 / 1111** | **1 / 1** |
| | Accuracy : 100% |

| Points Scored | Time Taken |
|---|---|
| **4 / 4** | **0.52** |
| Your Total Score: 26 ↑ | |

**Solve Next**

Large Factorial | Number following a pattern | Rank The Permutations

**Stay Ahead With:**

**Build 21 Projects in 21 Days**

Java (21) | Start Timer

```java
// User function Template for Java

class Solution {
    public static ArrayList<Integer> factorial(int n) {
        // code here
        ArrayList<Integer> result = new ArrayList<>();

        // 1! = 1
        result.add(1);

        for (int i = 2; i <= n; i++) {
            int carry = 0;

            for (int j = 0; j < result.size(); j++) {
                int val = result.get(j) * i + carry;
                result.set(j, val % 10);
                carry = val / 10;
            }

            while (carry > 0) {
                result.add(carry % 10);
                carry = carry / 10;
            }
        }

        // reverse because digits ulte store hue hain
        Collections.reverse(result);
        return result;
    }
}
```

Custom Input | Compile & Run | Submit

geeksforgeeks.org/problems/array-subset-of-another-array2317/1

Relaunch to update

Search...

Get 90% Refund

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

**1114 / 1114**

Attempts : Correct / Total

**1 / 1**

Accuracy : 100%

Points Scored

**1 / 1**

Your Total Score: 27

Time Taken

**0.56**

**Solve Next**

Counting elements in two arrays

Union of 2 Sorted Arrays | Left most and right most index

**Stay Ahead With:**

**Build 21 Projects in 21 Days**

Java (21)

Start Timer

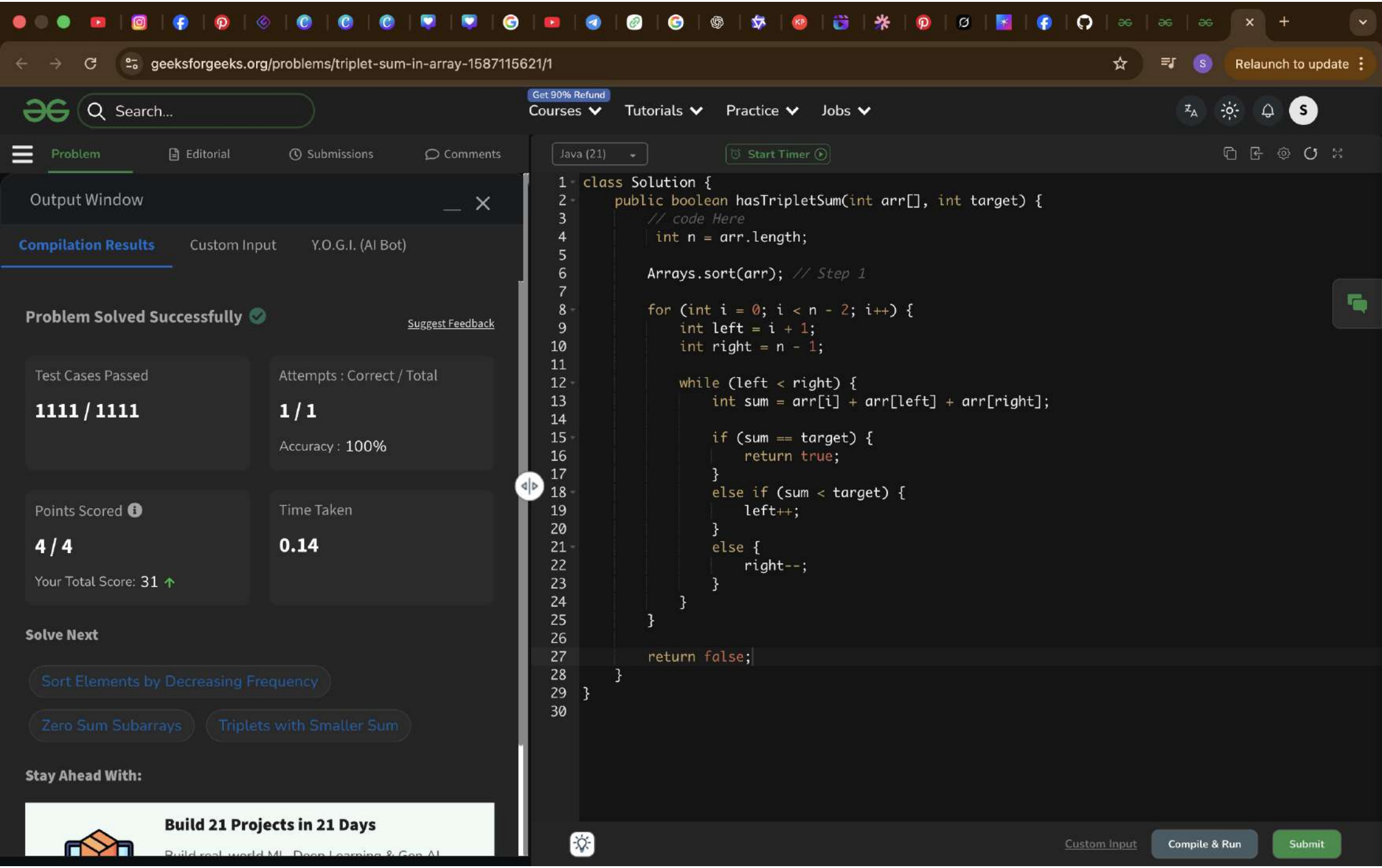
```java
class Solution {
    public boolean isSubset(int a[], int b[]) {
        // Your code here
        HashMap<Integer, Integer> map = new HashMap<>();

        // Frequency map of array a
        for (int num : a) {
            map.put(num, map.getOrDefault(num, 0) + 1);
        }

        // Check elements of array b
        for (int num : b) {
            if (!map.containsKey(num) || map.get(num) == 0) {
                return false;
            }
            map.put(num, map.get(num) - 1);
        }

        return true;
    }
}
```

Custom Input | Compile & Run | Submit


geeksforgeeks.org/problems/triplet-sum-in-array-1587115621/1
Relaunch to update
Search...
Get 90% Refund
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
Your Total Score: 31
Time Taken
0.14
Solve Next
Sort Elements by Decreasing Frequency
Zero Sum Subarrays
Triplets with Smaller Sum
Stay Ahead With:
Build 21 Projects in 21 Days
Java (21)
Start Timer
class Solution {
public boolean hasTripletSum(int arr[], int target) {
// code Here
int n = arr.length;
Arrays.sort(arr); // Step 1
for (int i = 0; i < n - 2; i++) {
int left = i + 1;
int right = n - 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;
}
}
Custom Input
Compile & Run
Submit

geeksforgeeks.org/problems/trapping-rain-water-1587115621/1

Relaunch to update

Search...

Get 90% Refund

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 | Attempts : Correct / Total |
|---|---|
| **1111 / 1111** | **1 / 1** |
| | Accuracy : 100% |

| Points Scored | Time Taken |
|---|---|
| **8 / 8** | **0.26** |
| Your Total Score: 39 ↑ | |

**Solve Next**

Longest Arithmetic Subsequence | Rod Cutting | Jump Game

**Stay Ahead With:**



Java (21)

Start Timer

```java
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;
    }
}
```

Custom Input | Compile & Run | Submit