

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 1 / 1 Time Taken 0.3

Your Total Score: 3 ↑

Solve Next

Type of array Largest in Array First and Second Smallests

Stay Ahead With:

Custom Input Compile & Run Submit

```
1 import java.util.ArrayList;
2
3 class Solution {
4     public ArrayList<Integer> getMinMax(int[] arr) {
5         int min = arr[0];
6         int max = arr[0];
7         for (int i = 1; i < arr.length; i++) {
8             if (arr[i] < min) {
9                 min = arr[i];
10            }
11            if (arr[i] > max) {
12                max = arr[i];
13            }
14        }
15        ArrayList<Integer> res = new ArrayList<>();
16        res.add(min);
17        res.add(max);
18
19        return res;
20    }
21 }
22 }
```

Search...

Courses Tutorials Practice Jobs

Problem Editorial Submissions Comments

Java (2.1) 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 2 / 2 Time Taken 0.78

Your Total Score: 2 ↑

Solve Next

Mountain Subarray Problem Java ArrayList Operation

Even and odd elements at even and odd positions

Stay Ahead With:

Custom Input Compile & Run Submit

```
1 class Solution {
2     public void reverseArray(int[] arr) {
3         int start = 0;
4         int end = arr.length - 1;
5
6         while (start < end) {
7
8             int temp = arr[start];
9             arr[start] = arr[end];
10            arr[end] = temp;
11
12            start++;
13            end--;
14        }
15    }
16}
```

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 1121 / 1121 Attempts : Correct / Total 1 / 1 Accuracy : 100%

Points Scored 4 / 4 Time Taken 0.69

Your Total Score: 7 ↑

Solve Next

Smallest Positive Missing Valid Pair Sum Optimal Array

Stay Ahead With:

Java (21) Start Timer

```
1 import java.util.Arrays;
2 import java.util.ArrayList;
3 class Solution
4 {
5     public int kthSmallest(int[] arr, int k)
6     {
7         Arrays.sort(arr);
8         return arr[k - 1];
9     }
10 }
```

Custom Input Compile & Run Submit Ctrl + Enter

Get 90% Refund Courses Tutorials Practice Jobs

Search... 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 2 / 2 Time Taken 0.96 Your Total Score: 9 ↑

Solve Next

Intersection of Arrays with Distinct LCM of given array elements

Perfect Squares in a Range

Java (21) Start Timer

```
1 import java.util.HashSet;
2 import java.util.ArrayList;
3 class Solution
4 {
5     public ArrayList<Integer> findUnion(int[] a, int[] b)
6     {
7         HashSet<Integer> set = new HashSet<>();
8         for (int num : a)
9         {
10             set.add(num);
11         }
12         for (int num : b)
13         {
14             set.add(num);
15         }
16         return new ArrayList<>(set);
17     }
18 }
```

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 0.9

Your Total Score: 10 ↑

Solve Next

Last index of One Pairs with Positive Negative values Repeated IDs

Stay Ahead With:

Custom Input Compile & Run Submit Ctrl + Enter

Java (21) Start Timer

```
1 class Solution
2 {
3     public int largest(int[] arr)
4     {
5         int max = arr[0];
6         for (int i = 1; i < arr.length; i++)
7         {
8             if (arr[i] > max)
9             {
10                 max = arr[i];
11             }
12         }
13     }
14     return max;
15 }
```

Get 90% Refund
Courses ▾ Tutorials ▾ Practice ▾ Jobs ▾

Search...     

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**

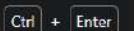
Your Total Score: 11 

Time Taken **0.97**

Solve Next

Third Largest Print an array in Pendulum Arrangement Inverse Permutation

Stay Ahead With:

Custom Input  Compile & Run Submit

```
Java (21) Start Timer
1 class Solution
2 {
3     public void rotate(int[] arr)
4     {
5         if (arr == null || arr.length <= 1)
6         {
7             return;
8         }
9         int last = arr[arr.length - 1];
10        for (int i = arr.length - 1; i > 0; i--)
11        {
12            arr[i] = arr[i - 1];
13        }
14        arr[0] = last;
15    }
16 }
```



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

1 / 1

Accuracy : 100%



Points Scored ⓘ

4 / 4

Time Taken

0.7

Your Total Score: 15 ↑

Solve Next

Count of Subarrays

Longest Arithmetic Subsequence

Smallest sum contiguous subarray

Java (21)

Start Timer

```
1 class Solution
2 {
3     public int maxSubarraySum(int[] arr)
4     {
5         int maxSum = arr[0];
6         int currSum = arr[0];
7         for (int i = 1; i < arr.length; i++)
8         {
9             currSum = Math.max(arr[i], currSum + arr[i]);
10            maxSum = Math.max(maxSum, currSum);
11        }
12        return maxSum;
13    }
14 }
15 }
```

Custom Input

Compile & Run

Submit

Array < > ✎

Submit

Premium

Description Accepted Editorial Solutions Submissions

All Submissions

Accepted 66 / 66 testcases passed

vt2812 submitted at Feb 12, 2026 23:33

Runtime 0 ms | Beats 100.00% 🏆

Memory 45.01 MB | Beats 10.18%

Analyze Complexity

Runtime distribution chart showing a single bar at 0ms.

Code

Java Auto

```
1 class Solution {  
2     public int searchInsert(int[] nums, int target) {  
3         int low = 0;  
4         int high = nums.length - 1;  
5  
6         while (low <= high) {  
7             int mid = low + (high - low) / 2;  
8  
9             if (nums[mid] == target) {  
10                 return mid;  
11             } else if (nums[mid] < target) {  
12                 low = mid + 1;  
13             } else {  
14                 high = mid - 1;  
15             }  
16         }  
17  
18         return low;  
19     }  
20 }
```

Saved Ln 20, Col

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

```
1 class Solution {  
2     public int searchInsert(int[] nums, int target) {  
3         int low = 0;  
4         int high = nums.length - 1;  
5  
6         while (low <= high) {
```

Array < > ⌂

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 63 / 63 testcases passed

vt2812 submitted at Feb 12, 2026 23:36

Runtime 2 ms | Beats 99.17% Analyze Complexity

Memory 47.34 MB | Beats 20.09%

Code

Java Auto

```
1 import java.util.HashMap;
2 import java.util.Map;
3
4 class Solution {
5     public int[] twoSum(int[] nums, int target) {
6
7         Map<Integer, Integer> map = new HashMap<>();
8
9         for (int i = 0; i < nums.length; i++) {
10             int complement = target - nums[i];
11             if (map.containsKey(complement)) {
12                 return new int[] { map.get(complement), i };
13             }
14
15             map.put(nums[i], i);
16         }
17     }
18 }
19 }
```

Saved In 19, Col 2

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

Get 90% Refund

Courses ▾ Tutorials ▾ Practice ▾ Jobs ▾

Search...

Problem Editorial Submissions Comments

Output Window

Java (21) Start Timer

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

Problem Solved Successfully

Suggest Feedback

Attempts : Correct / Total
1 / 1

Accuracy : 100%

Test Cases Passed
1120 / 1120

Time Taken
0.56

Points Scored
4 / 4

Your Total Score: **19**

Solve Next

Minimize the Heights II Jump Game Wine Buying and Selling

Stay Ahead With:

Custom Input Compile & Run Submit

```
1 class Solution
2 {
3     public int minJumps(int[] arr)
4     {
5         int n = arr.length;
6         if (arr[0] == 0) return -1;
7         int jumps = 0;
8         int farthest = 0;
9         int currentEnd = 0;
10        for (int i = 0; i < n - 1; i++)
11        {
12            farthest = Math.max(farthest, i + arr[i]);
13            if (i == currentEnd) {
14                jumps++;
15                currentEnd = farthest;
16                if (currentEnd <= i) return -1;
17                if (currentEnd >= n - 1) return jumps;
18            }
19        }
20        return (currentEnd >= n - 1) ? jumps : -1;
21    }
22 }
23 }
```