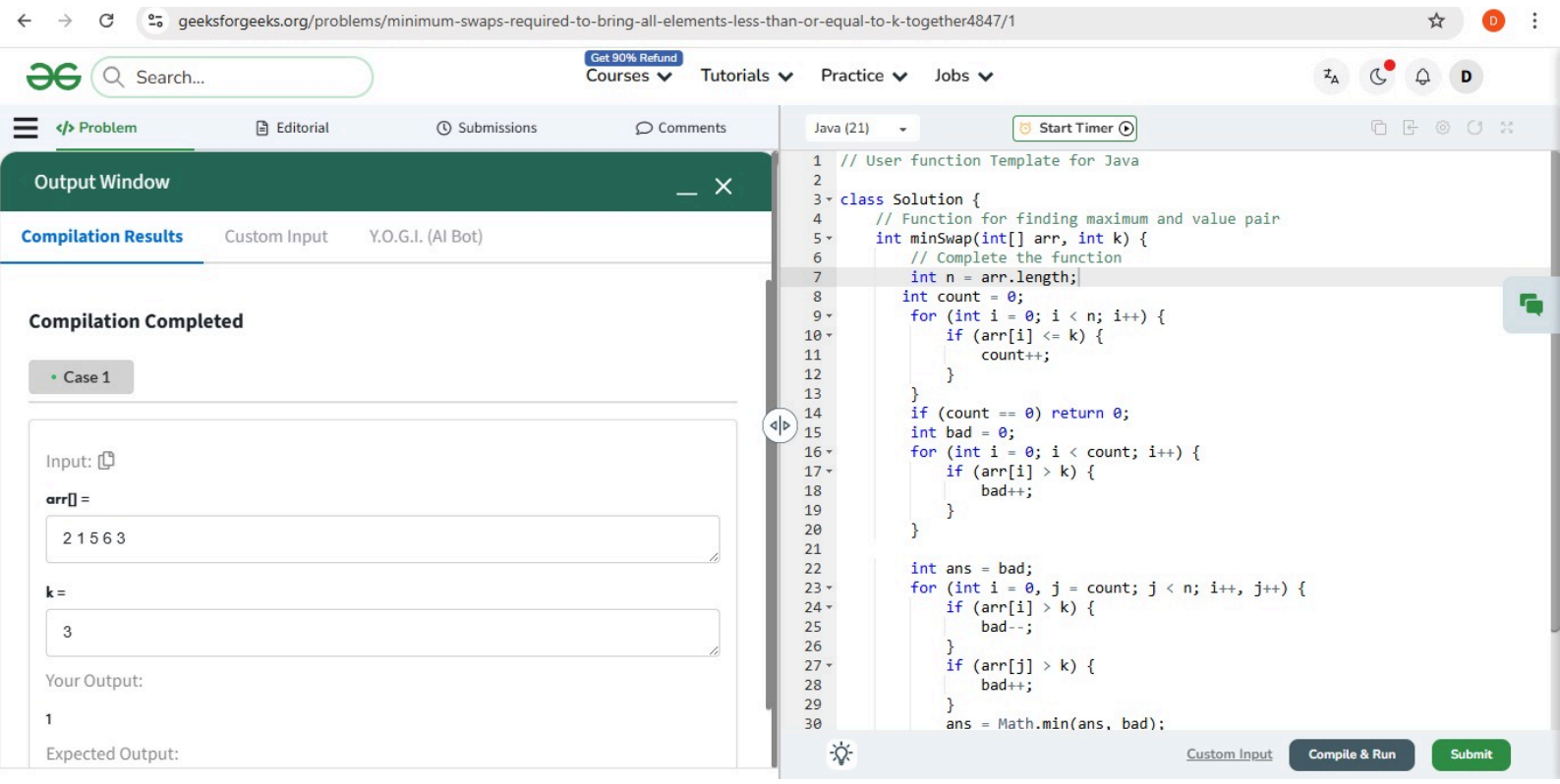
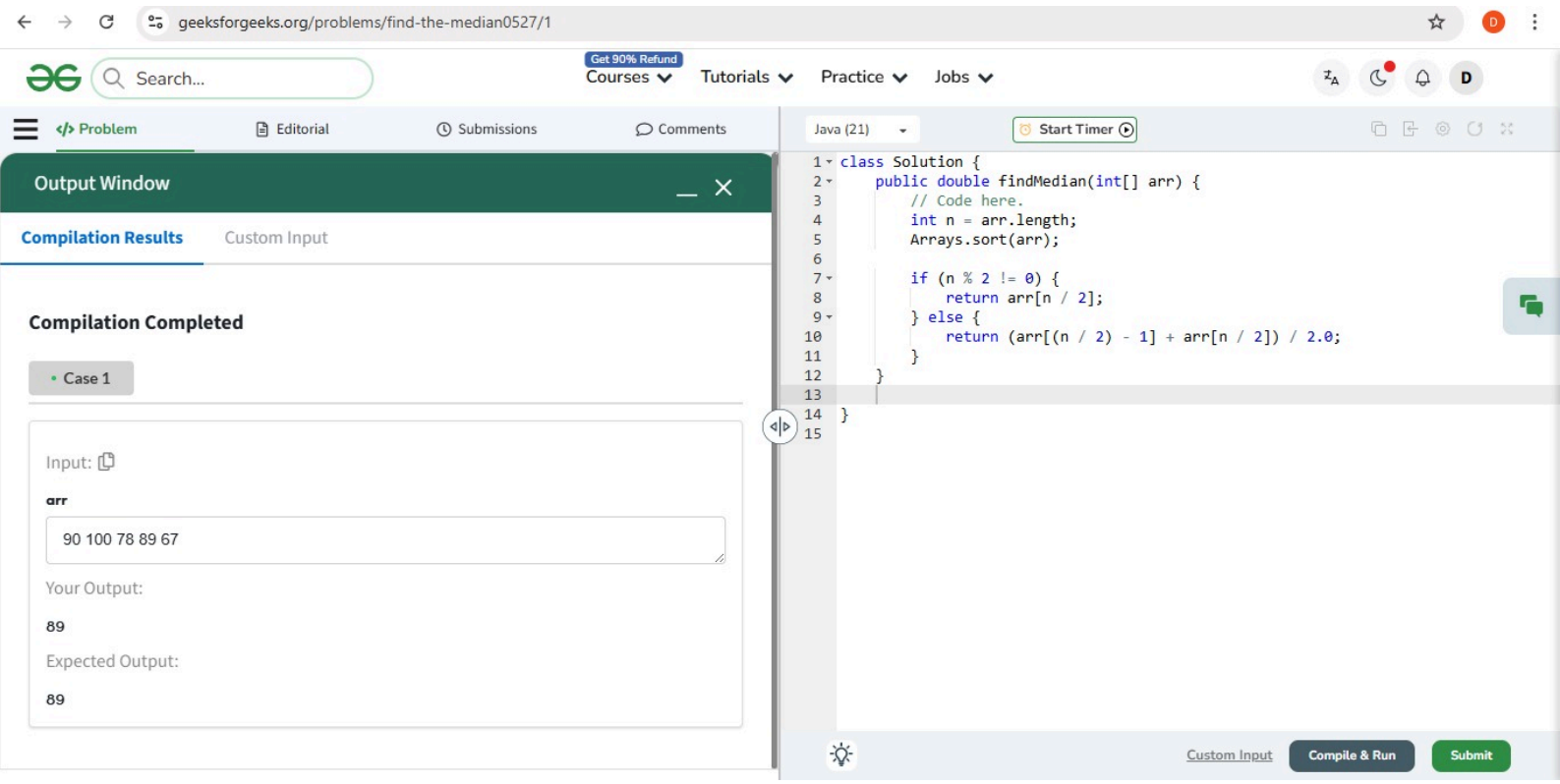

geeksforgeeks.org/problems/minimum-swaps-required-to-bring-all-elements-less-than-or-equal-to-k-together4847/1
Search...
Get 90% Refund
Courses
Tutorials
Practice
Jobs
Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Compilation Completed
Case 1
Input:
arr[] =
2 1 5 6 3
k =
3
Your Output:
1
Expected Output:
Java (21)
Start Timer
1 // User function Template for Java
2
3 class Solution {
4 // Function for finding maximum and value pair
5 int minSwap(int[] arr, int k) {
6 // Complete the function
7 int n = arr.length;
8 int count = 0;
9 for (int i = 0; i < n; i++) {
10 if (arr[i] <= k) {
11 count++;
12 }
13 }
14 if (count == 0) return 0;
15 int bad = 0;
16 for (int i = 0; i < count; i++) {
17 if (arr[i] > k) {
18 bad++;
19 }
20 }
21
22 int ans = bad;
23 for (int i = 0, j = count; j < n; i++, j++) {
24 if (arr[i] > k) {
25 bad--;
26 }
27 if (arr[j] > k) {
28 bad++;
29 }
30 ans = Math.min(ans, bad);
Custom Input
Compile & Run
Submit


geeksforgeeks.org/problems/find-the-median0527/1
Search...
Get 90% Refund
Courses
Tutorials
Practice
Jobs
Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Compilation Completed
Case 1
Input:
arr
90 100 78 89 67
Your Output:
89
Expected Output:
89
Java (21)
Start Timer
1 class Solution {
2 public double findMedian(int[] arr) {
3 // Code here.
4 int n = arr.length;
5 Arrays.sort(arr);
6
7 if (n % 2 != 0) {
8 return arr[n / 2];
9 } else {
10 return (arr[(n / 2) - 1] + arr[n / 2]) / 2.0;
11 }
12 }
13
14 }
15
Custom Input
Compile & Run
Submit

geeksforgeeks.org/problems/spirally-traversing-a-matrix-1587115621/1

Search... Get 90% Refund Courses Tutorials Practice Jobs D

Problem Editorial Submissions Comments

Spirally traversing a matrix

Difficulty: **Medium** Accuracy: **35.2%** Submissions: **342K+** Points: **4**

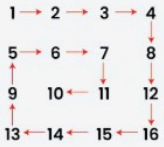
You are given a rectangular matrix **mat[][]** of size **n x m**, and your task is to return an array while **traversing** the matrix in **spiral** form.

Examples:

Input: mat[][] = [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12], [13, 14, 15, 16]]
Output: [1, 2, 3, 4, 8, 12, 16, 15, 14, 13, 9, 5, 6, 7, 11, 10]
Explanation:

Example of matrix in spiral form

Matrix:

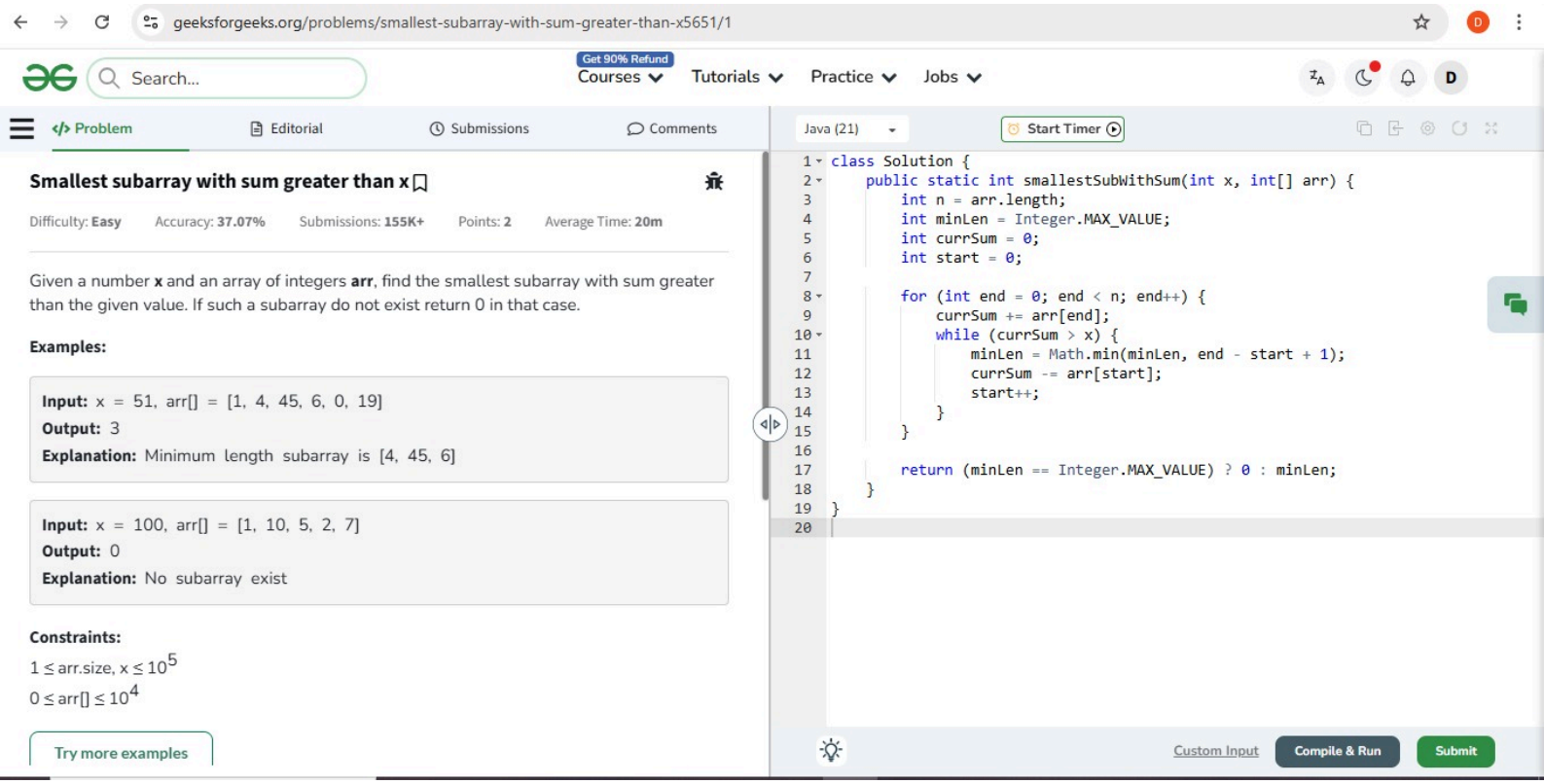


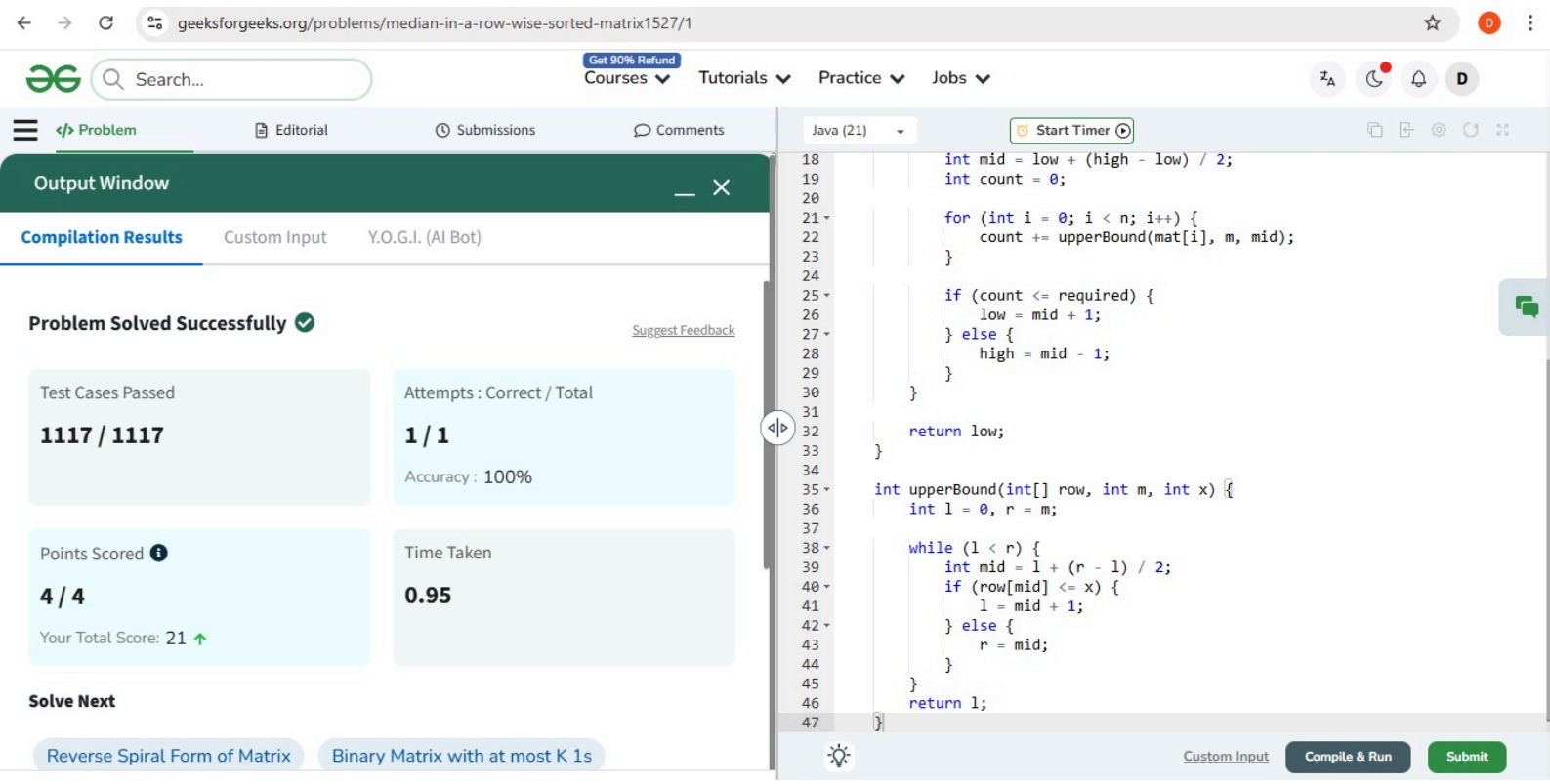
Output: 1, 2, 3, 4, 8, 12, 16, 15, 14, 13, 9, 5, 6, 7, 11, 10

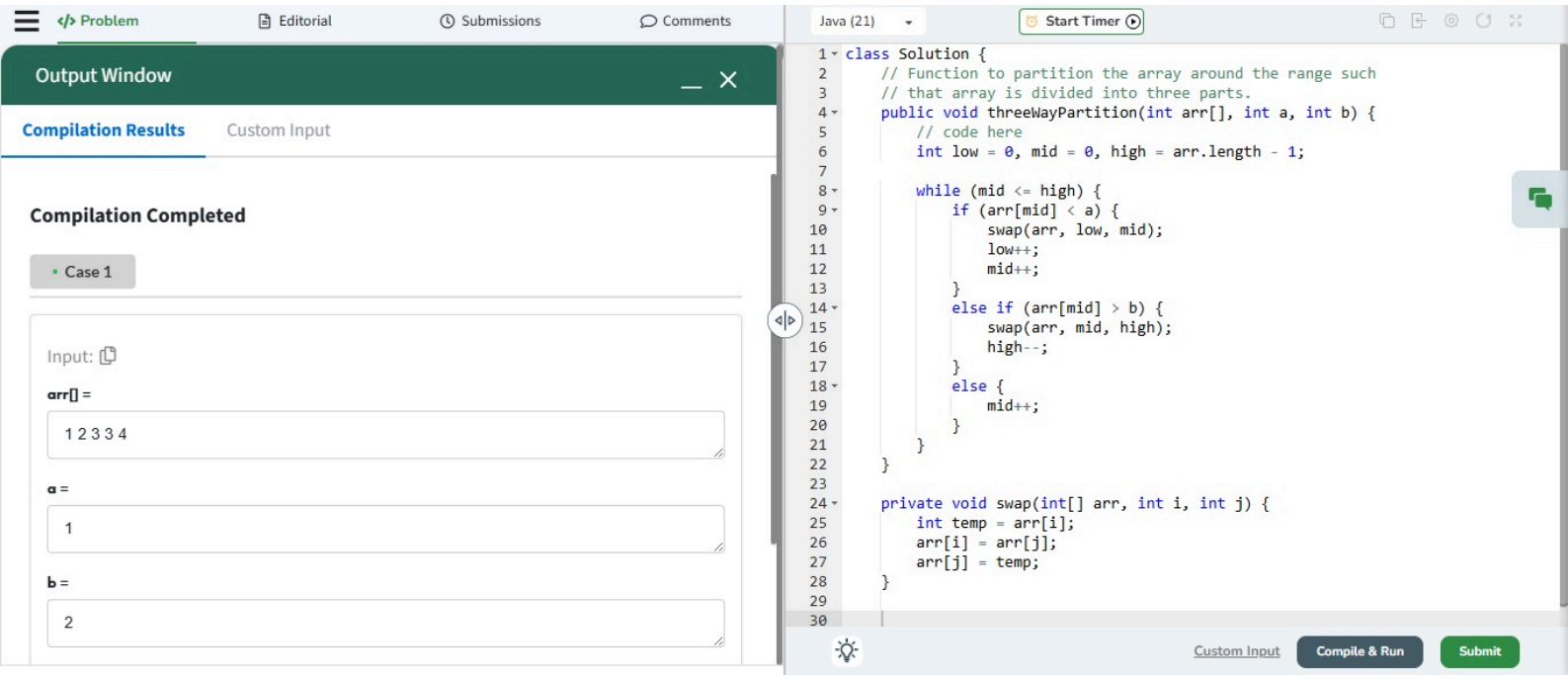
Java (21) Start Timer

```java
class Solution {
    public ArrayList<Integer> spirallyTraverse(int[][] mat) {
        // code here
        ArrayList<Integer> res = new ArrayList<>();

        int n = mat.length;
        int m = mat[0].length;

        int top = 0, bottom = n - 1;
        int left = 0, right = m - 1;

        while (top <= bottom && left <= right) {

            for (int i = left; i <= right; i++) {
                res.add(mat[top][i]);
            }
            top++;

            for (int i = top; i <= bottom; i++) {
                res.add(mat[i][right]);
            }
            right--;

            if (top <= bottom) {
                for (int i = right; i >= left; i--) {
                    res.add(mat[bottom][i]);
                }
                bottom--;
            }
```

Custom Input Compile & Run Submit


geeksforgeeks.org/problems/smallest-subarray-with-sum-greater-than-x5651/1
Search...
Get 90% Refund
Courses
Tutorials
Practice
Jobs
Problem
Editorial
Submissions
Comments
Java (21)
Start Timer
Smallest subarray with sum greater than x
Difficulty: Easy
Accuracy: 37.07%
Submissions: 155K+
Points: 2
Average Time: 20m
Given a number x and an array of integers arr, find the smallest subarray with sum greater than the given value. If such a subarray do not exist return 0 in that case.
Examples:
Input: x = 51, arr[] = [1, 4, 45, 6, 0, 19]
Output: 3
Explanation: Minimum length subarray is [4, 45, 6]
Input: x = 100, arr[] = [1, 10, 5, 2, 7]
Output: 0
Explanation: No subarray exist
Constraints:
1 ≤ arr.size, x ≤ 10^5
0 ≤ arr[] ≤ 10^4
Try more examples
class Solution {
public static int smallestSubWithSum(int x, int[] arr) {
int n = arr.length;
int minLen = Integer.MAX_VALUE;
int currSum = 0;
int start = 0;
for (int end = 0; end < n; end++) {
currSum += arr[end];
while (currSum > x) {
minLen = Math.min(minLen, end - start + 1);
currSum -= arr[start];
start++;
}
}
return (minLen == Integer.MAX_VALUE) ? 0 : minLen;
}
}
Custom Input
Compile & Run
Submit


geeksforgeeks.org/problems/median-in-a-row-wise-sorted-matrix1527/1
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
1117 / 1117
Attempts : Correct / Total
1 / 1
Accuracy : 100%
Points Scored
4 / 4
Your Total Score: 21
Time Taken
0.95
Solve Next
Reverse Spiral Form of Matrix
Binary Matrix with at most K 1s
int mid = low + (high - low) / 2;
int count = 0;
for (int i = 0; i < n; i++) {
count += upperBound(mat[i], m, mid);
}
if (count <= required) {
low = mid + 1;
} else {
high = mid - 1;
}
}
return low;
}
int upperBound(int[] row, int m, int x) {
int l = 0, r = m;
while (l < r) {
int mid = l + (r - l) / 2;
if (row[mid] <= x) {
l = mid + 1;
} else {
r = mid;
}
}
return l;
}
Custom Input
Compile & Run
Submit


Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Compilation Completed
Case 1
Input:
arr[] =
1 2 3 3 4
a =
1
b =
2
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 low = 0, mid = 0, high = arr.length - 1;
while (mid <= high) {
if (arr[mid] < a) {
swap(arr, low, mid);
low++;
mid++;
}
else if (arr[mid] > b) {
swap(arr, mid, high);
high--;
}
else {
mid++;
}
}
}
private void swap(int[] arr, int i, int j) {
int temp = arr[i];
arr[i] = arr[j];
arr[j] = temp;
}
Custom Input
Compile & Run
Submit