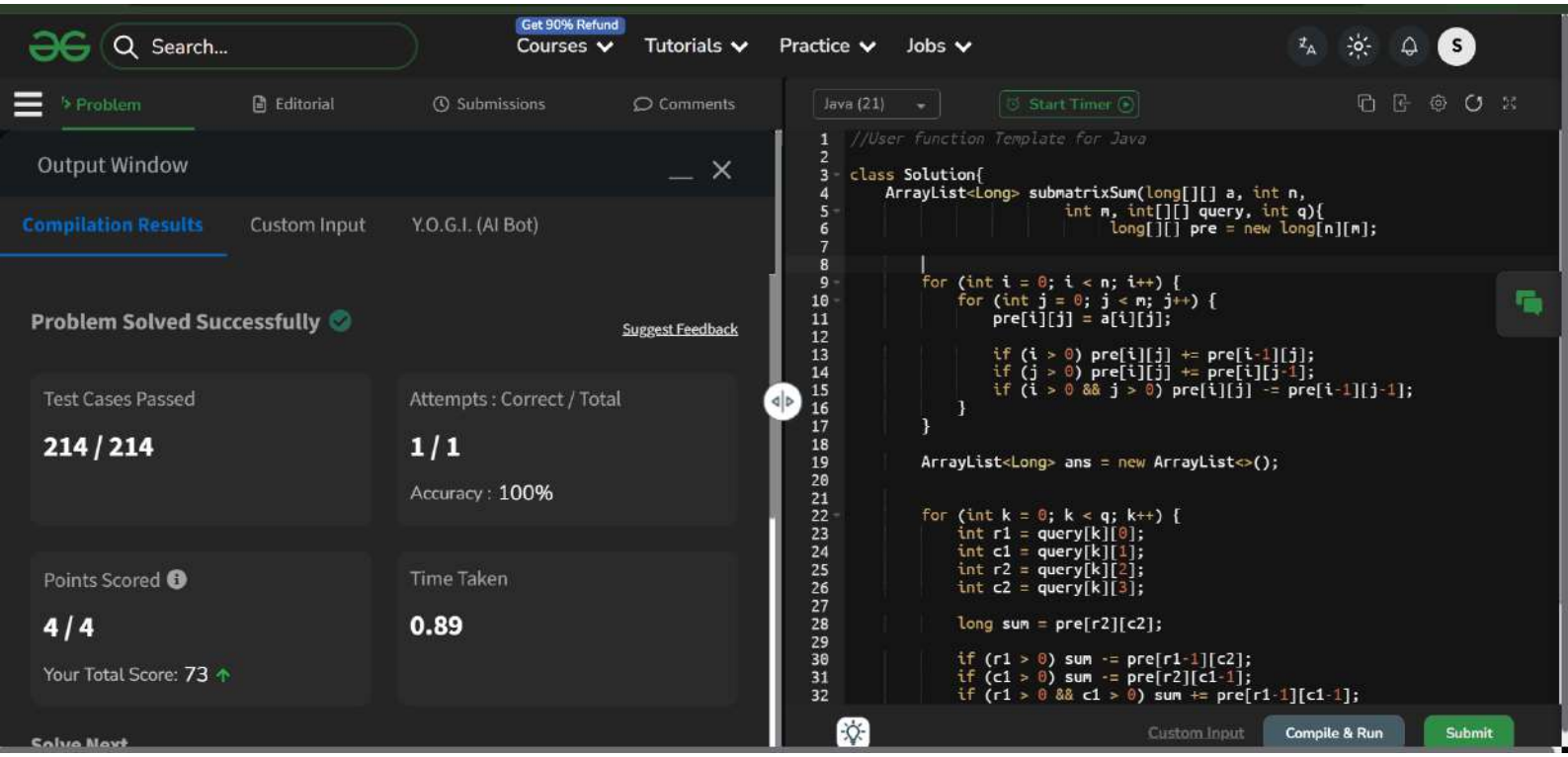
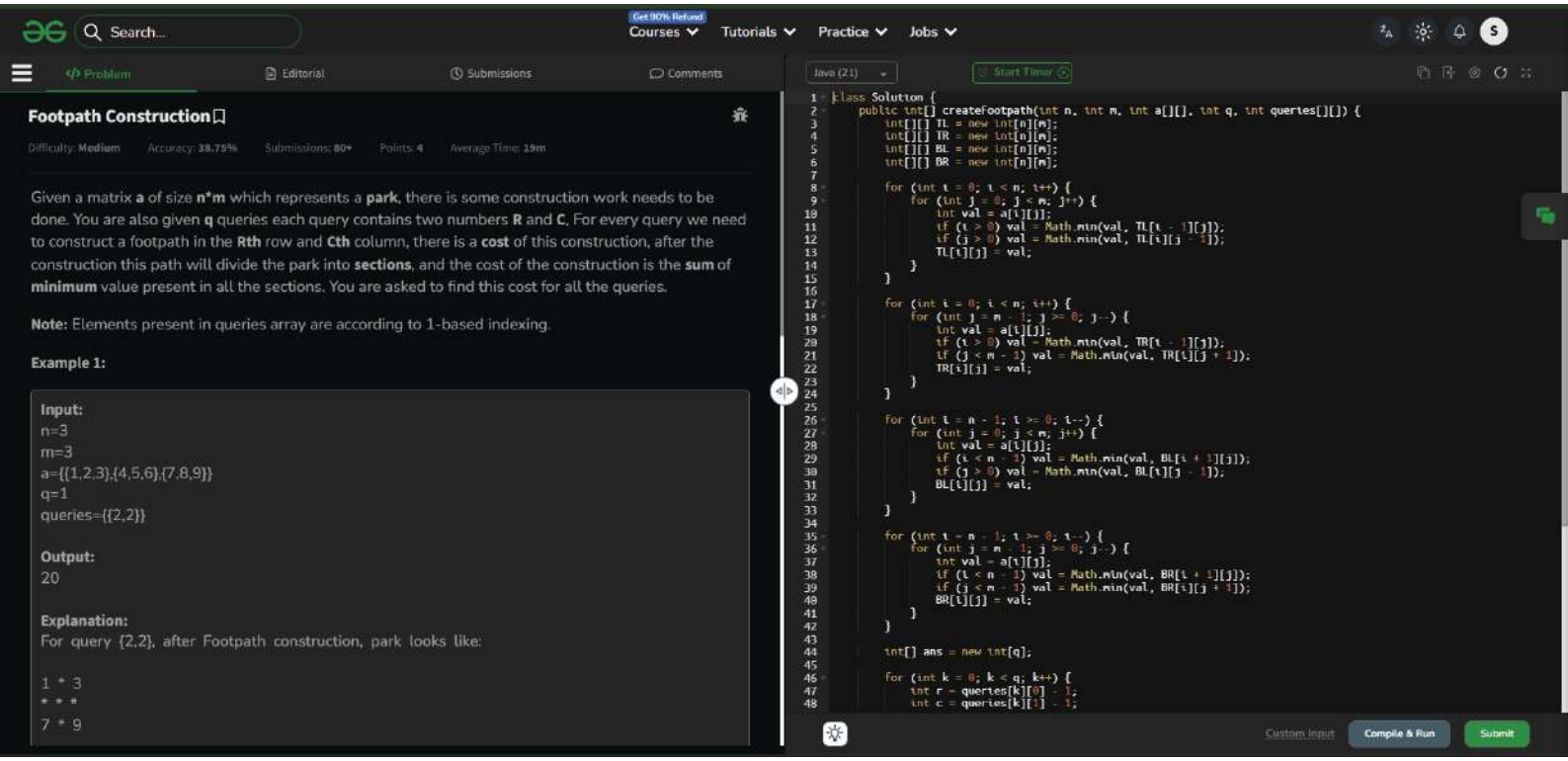
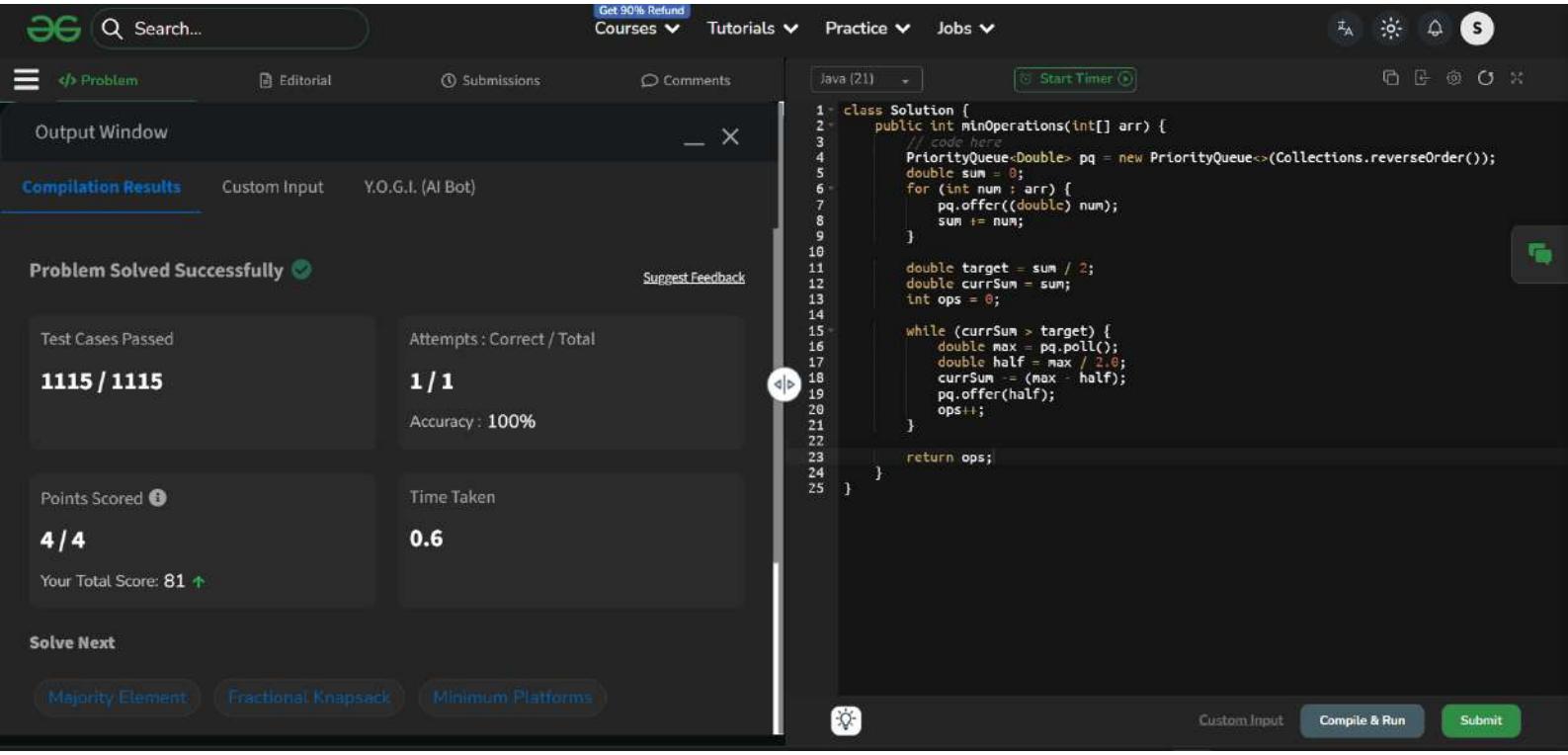
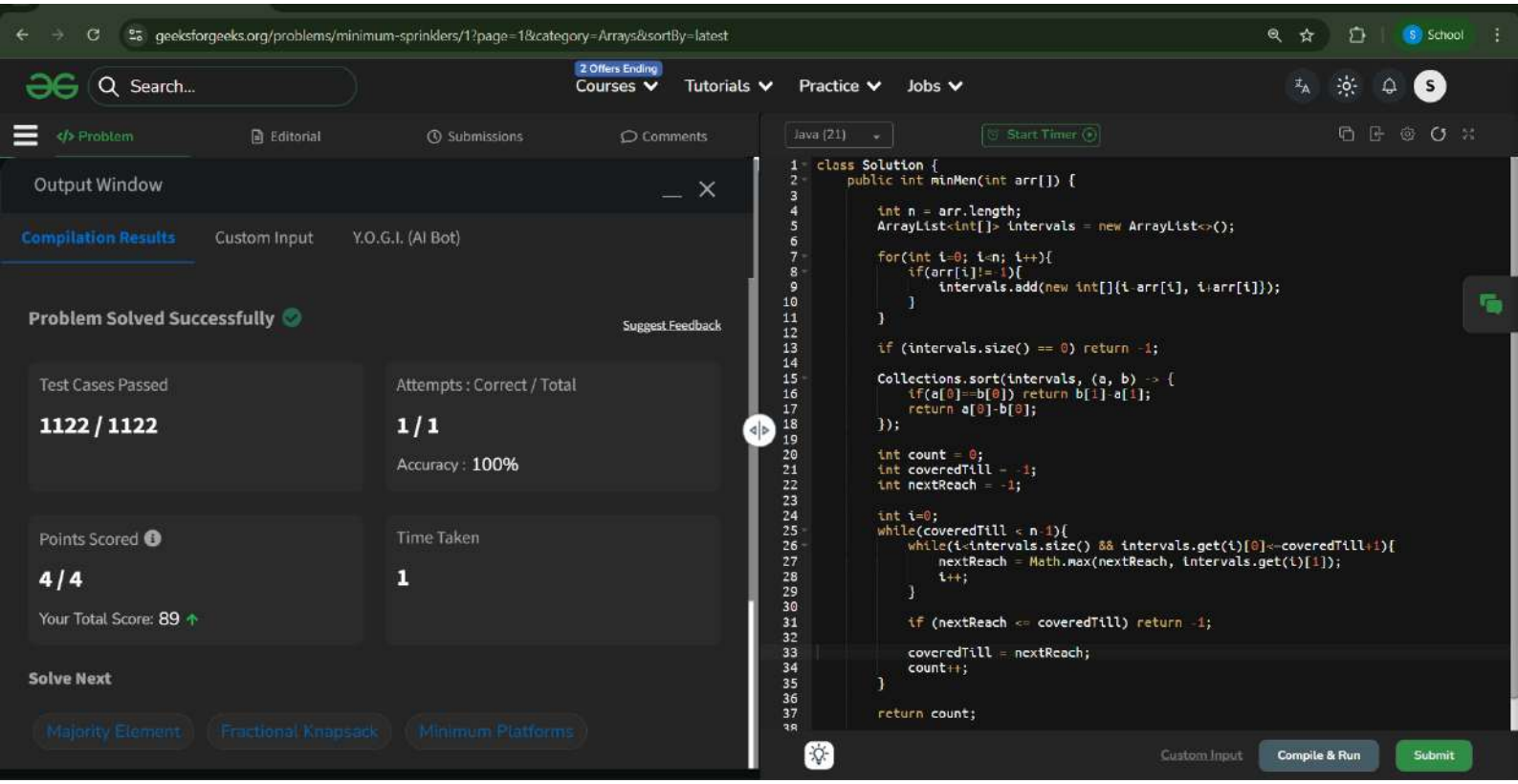
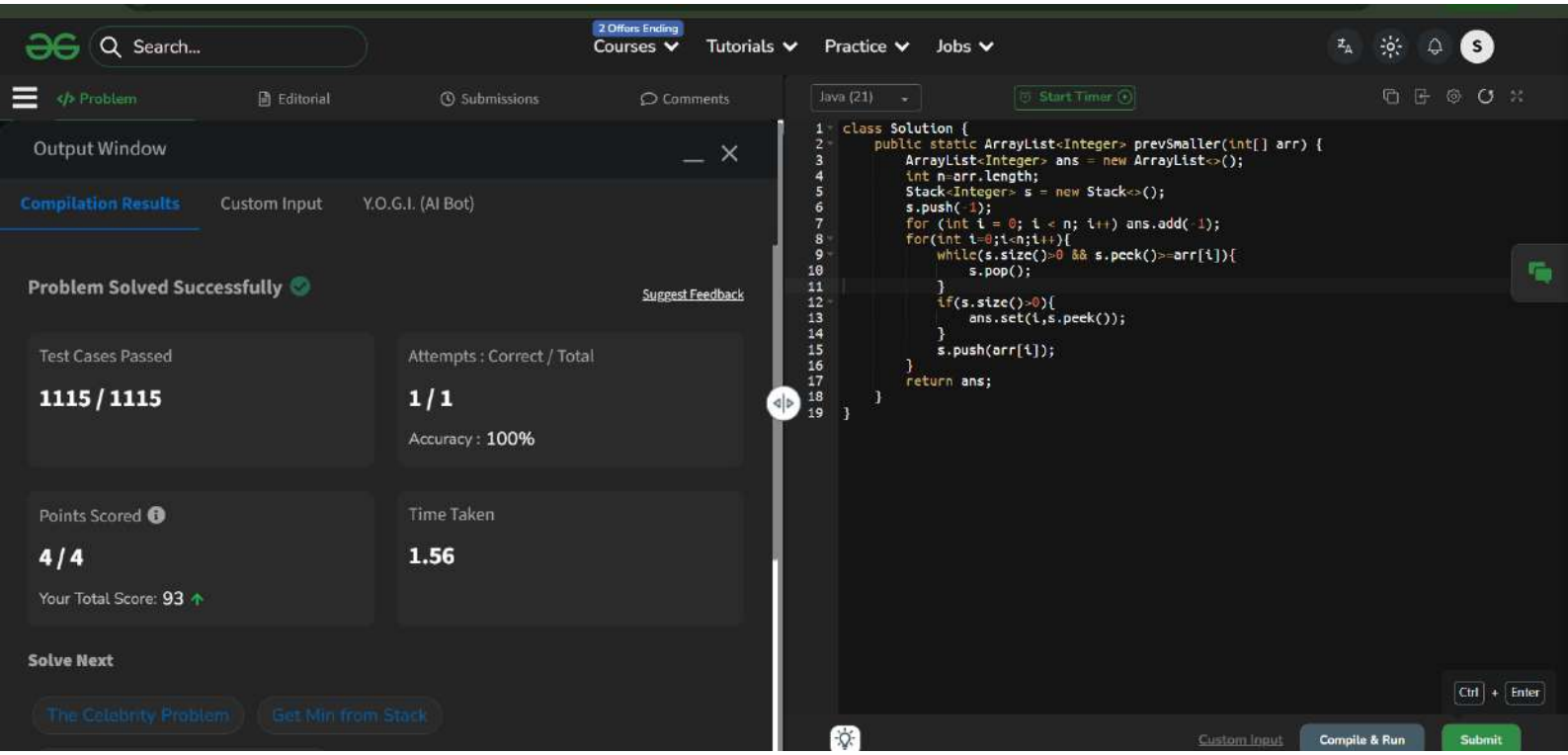
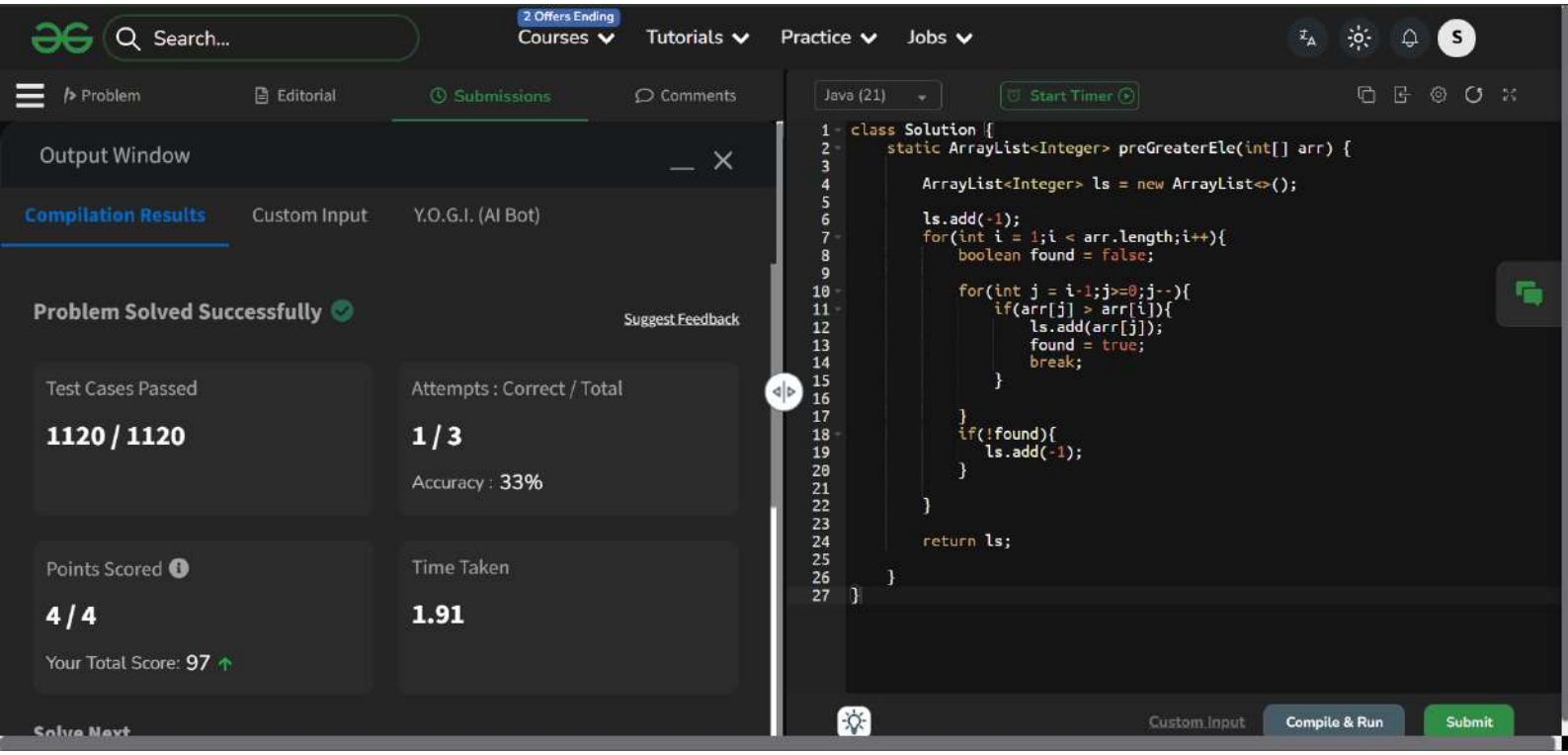

Search...
Get 90% Refund
Courses
Tutorials
Practice
Jobs
Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
214 / 214
Attempts : Correct / Total
1 / 1
Accuracy : 100%
Points Scored
4 / 4
Your Total Score: 73
Time Taken
0.89
Solve Next
Java (21)
Start Timer
//User function Template for Java
class Solution{
ArrayList<Long> submatrixSum(long[][] a, int n,
int m, int[][] query, int q){
long[][] pre = new long[n][m];
for (int i = 0; i < n; i++) {
for (int j = 0; j < m; j++) {
pre[i][j] = a[i][j];
if (i > 0) pre[i][j] += pre[i-1][j];
if (j > 0) pre[i][j] += pre[i][j-1];
if (i > 0 && j > 0) pre[i][j] -= pre[i-1][j-1];
}
}
ArrayList<Long> ans = new ArrayList<>();
for (int k = 0; k < q; k++) {
int r1 = query[k][0];
int c1 = query[k][1];
int r2 = query[k][2];
int c2 = query[k][3];
long sum = pre[r2][c2];
if (r1 > 0) sum -= pre[r1-1][c2];
if (c1 > 0) sum -= pre[r2][c1-1];
if (r1 > 0 && c1 > 0) sum += pre[r1-1][c1-1];
Custom Input
Compile & Run
Submit


Search...
Get 90% Refund
Courses
Tutorials
Practice
Jobs
Problem
Editorial
Submissions
Comments
Java (21)
Start Timer
Footpath Construction
Difficulty: Medium
Accuracy: 38.75%
Submissions: 80+
Points: 4
Average Time: 19m
Given a matrix a of size n*m which represents a park, there is some construction work needs to be done. You are also given q queries each query contains two numbers R and C. For every query we need to construct a footpath in the Rth row and Cth column, there is a cost of this construction, after the construction this path will divide the park into sections, and the cost of the construction is the sum of minimum value present in all the sections. You are asked to find this cost for all the queries.
Note: Elements present in queries array are according to 1-based indexing.
Example 1:
Input:
n=3
m=3
a={{1,2,3},{4,5,6},{7,8,9}}
q=1
queries={{2,2}}
Output:
20
Explanation:
For query {2,2}, after Footpath construction, park looks like:
1 * 3
* * *
7 * 9
class Solution {
public int[] createFootpath(int n, int m, int a[][], int q, int queries[][]) {
int[][] TL = new int[n][m];
int[][] TR = new int[n][m];
int[][] BL = new int[n][m];
int[][] BR = new int[n][m];
for (int i = 0; i < n; i++) {
for (int j = 0; j < m; j++) {
int val = a[i][j];
if (i > 0) val = Math.min(val, TL[i - 1][j]);
if (j > 0) val = Math.min(val, TL[i][j - 1]);
TL[i][j] = val;
}
}
for (int i = 0; i < n; i++) {
for (int j = m - 1; j >= 0; j--) {
int val = a[i][j];
if (i > 0) val = Math.min(val, TR[i - 1][j]);
if (j < m - 1) val = Math.min(val, TR[i][j + 1]);
TR[i][j] = val;
}
}
for (int i = n - 1; i >= 0; i--) {
for (int j = 0; j < m; j++) {
int val = a[i][j];
if (i < n - 1) val = Math.min(val, BL[i + 1][j]);
if (j > 0) val = Math.min(val, BL[i][j - 1]);
BL[i][j] = val;
}
}
for (int i = n - 1; i >= 0; i--) {
for (int j = m - 1; j >= 0; j--) {
int val = a[i][j];
if (i < n - 1) val = Math.min(val, BR[i + 1][j]);
if (j < m - 1) val = Math.min(val, BR[i][j + 1]);
BR[i][j] = val;
}
}
int[] ans = new int[q];
for (int k = 0; k < q; k++) {
int r = queries[k][0] - 1;
int c = queries[k][1] - 1;
Custom Input
Compile & Run
Submit


Search...
Get 90% Refund
Courses
Tutorials
Practice
Jobs
S
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
4 / 4
Your Total Score: 81
Time Taken
0.6
Solve Next
Majority Element
Fractional Knapsack
Minimum Platforms
Java (21)
Start Timer
1 class Solution {
2 public int minOperations(int[] arr) {
3 // code here
4 PriorityQueue<Double> pq = new PriorityQueue<>(Collections.reverseOrder());
5 double sum = 0;
6 for (int num : arr) {
7 pq.offer((double) num);
8 sum += num;
9 }
10
11 double target = sum / 2;
12 double currSum = sum;
13 int ops = 0;
14
15 while (currSum > target) {
16 double max = pq.poll();
17 double half = max / 2.0;
18 currSum -= (max - half);
19 pq.offer(half);
20 ops++;
21 }
22
23 return ops;
24 }
25 }
Custom Input
Compile & Run
Submit


geeksforgeeks.org/problems/minimum-sprinklers/1?page=1&category=Arrays&sortBy=latest
School
Search...
2 Offers Ending
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
1122 / 1122
Attempts : Correct / Total
1 / 1
Accuracy : 100%
Points Scored
4 / 4
Your Total Score: 89
Time Taken
1
Solve Next
Majority Element
Fractional Knapsack
Minimum Platforms
Java (21)
Start Timer
class Solution {
public int minMen(int arr[]) {
int n = arr.length;
ArrayList<int[]> intervals = new ArrayList<>();
for(int i=0; i<n; i++){
if(arr[i]!=-1){
intervals.add(new int[]{i-arr[i], i+arr[i]});
}
}
if (intervals.size() == 0) return -1;
Collections.sort(intervals, (a, b) -> {
if(a[0]==b[0]) return b[1]-a[1];
return a[0]-b[0];
});
int count = 0;
int coveredTill = -1;
int nextReach = -1;
int i=0;
while(coveredTill < n-1){
while(i<intervals.size() && intervals.get(i)[0]<=coveredTill+1){
nextReach = Math.max(nextReach, intervals.get(i)[1]);
i++;
}
if (nextReach <= coveredTill) return -1;
coveredTill = nextReach;
count++;
}
return count;
Custom Input
Compile & Run
Submit


Search...
2 Offers Ending
Courses
Tutorials
Practice
Jobs
S
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
4 / 4
Your Total Score: 93
Time Taken
1.56
Solve Next
The Celebrity Problem
Get Min from Stack
Java (21)
Start Timer
class Solution {
public static ArrayList<Integer> prevSmaller(int[] arr) {
ArrayList<Integer> ans = new ArrayList<>();
int n=arr.length;
Stack<Integer> s = new Stack<>();
s.push(-1);
for (int i = 0; i < n; i++) ans.add(-1);
for(int i=0;i<n;i++){
while(s.size()>0 && s.peek()>=arr[i]){
s.pop();
}
if(s.size()>0){
ans.set(i,s.peek());
}
s.push(arr[i]);
}
return ans;
}
}
Ctrl + Enter
Custom Input
Compile & Run
Submit


Search...
2 Offers Ending
Courses
Tutorials
Practice
Jobs
S
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 / 3
Accuracy : 33%
Points Scored
4 / 4
Your Total Score: 97
Time Taken
1.91
Solve Next
Java (21)
Start Timer
class Solution {
static ArrayList<Integer> preGreaterEle(int[] arr) {
ArrayList<Integer> ls = new ArrayList<>();
ls.add(-1);
for(int i = 1;i < arr.length;i++){
boolean found = false;
for(int j = i-1;j>=0;j--){
if(arr[j] > arr[i]){
ls.add(arr[j]);
found = true;
break;
}
}
if(!found){
ls.add(-1);
}
}
return ls;
}
}
Custom Input
Compile & Run
Submit