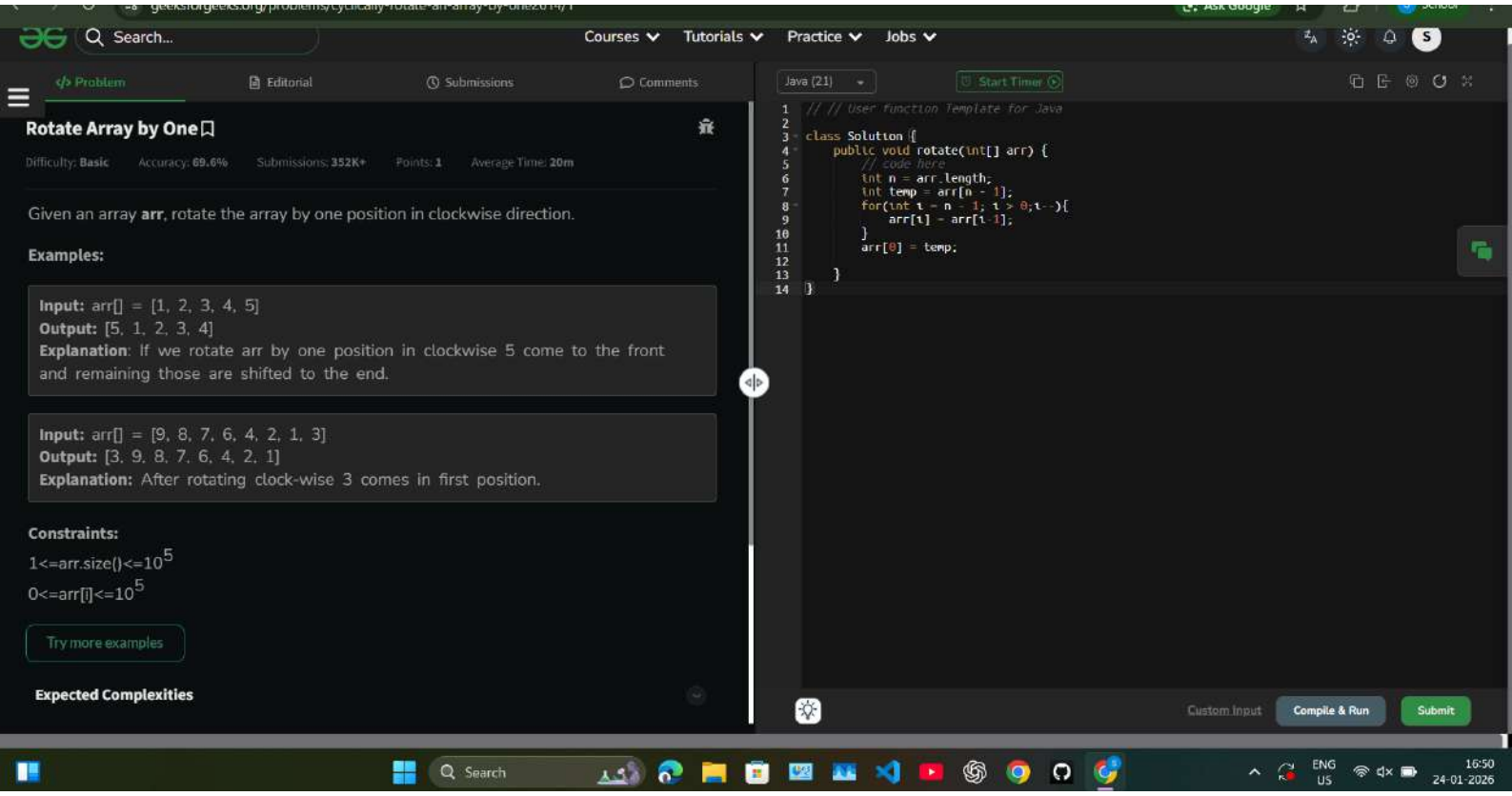
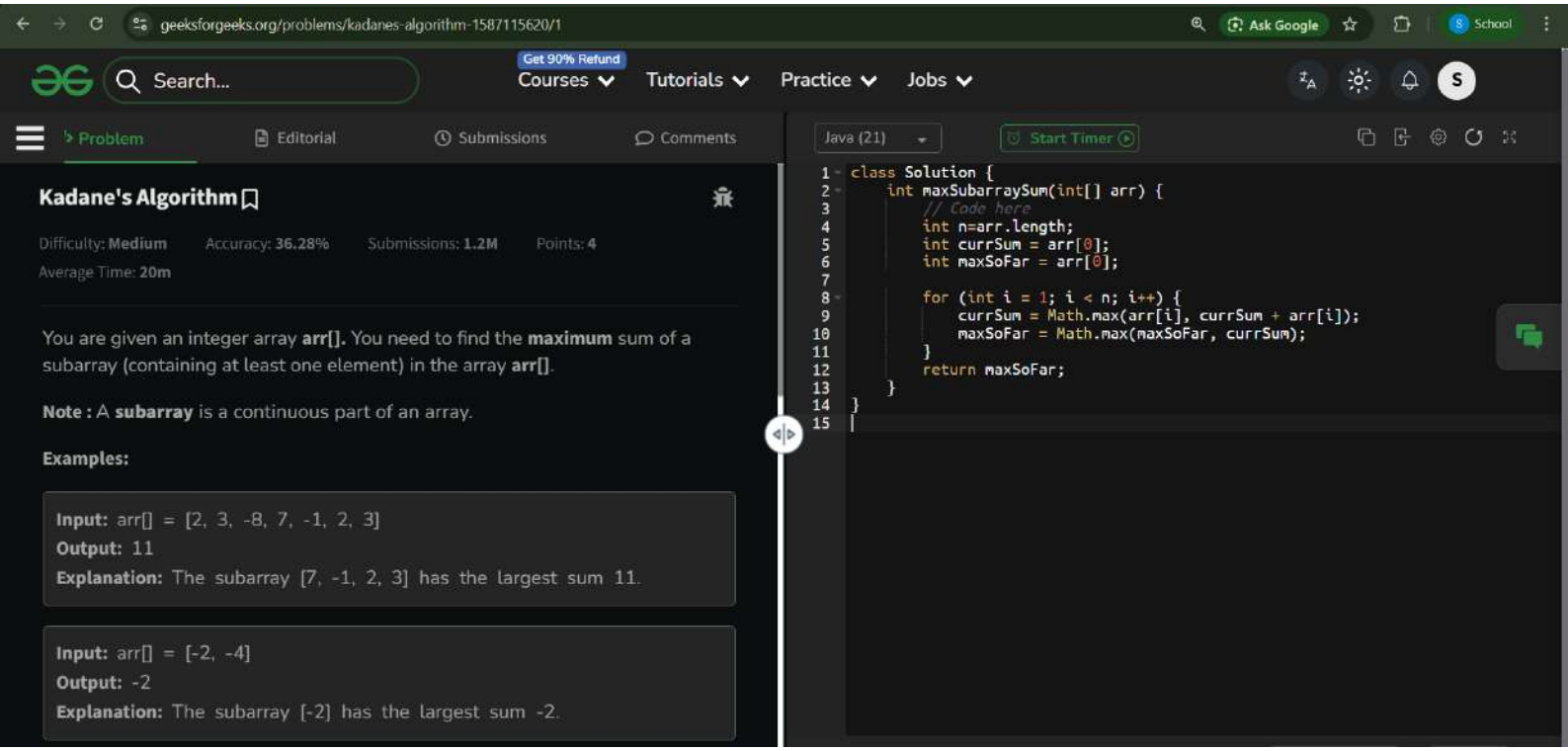

Search...
Courses
Tutorials
Practice
Jobs
Problem
Editorial
Submissions
Comments
Rotate Array by One
Difficulty: Basic
Accuracy: 69.6%
Submissions: 352K+
Points: 1
Average Time: 20m
Given an array arr, rotate the array by one position in clockwise direction.
Examples:
Input: arr[] = [1, 2, 3, 4, 5]
Output: [5, 1, 2, 3, 4]
Explanation: If we rotate arr by one position in clockwise 5 come to the front and remaining those are shifted to the end.
Input: arr[] = [9, 8, 7, 6, 4, 2, 1, 3]
Output: [3, 9, 8, 7, 6, 4, 2, 1]
Explanation: After rotating clock-wise 3 comes in first position.
Constraints:
1<=arr.size()<=10^5
0<=arr[i]<=10^5
Try more examples
Expected Complexities
Java (21)
Start Timer
// // User function Template for Java
class Solution {
public void rotate(int[] arr) {
// code here
int n = arr.length;
int temp = arr[n - 1];
for(int i = n - 1; i > 0;i--){
arr[i] = arr[i-1];
}
arr[0] = temp;
}
}
Custom Input
Compile & Run
Submit
Search
ENG
US
16:50
24-01-2026


geeksforgeeks.org/problems/kadanes-algorithm-1587115620/1
Ask Google
School
Search...
Get 90% Refund
Courses
Tutorials
Practice
Jobs
S
Problem
Editorial
Submissions
Comments
Java (21)
Start Timer
Kadane's Algorithm
Difficulty: Medium
Accuracy: 36.28%
Submissions: 1.2M
Points: 4
Average Time: 20m
You are given an integer array arr[]. You need to find the maximum sum of a subarray (containing at least one element) in the array arr[].
Note : A subarray is a continuous part of an array.
Examples:
Input: arr[] = [2, 3, -8, 7, -1, 2, 3]
Output: 11
Explanation: The subarray [7, -1, 2, 3] has the largest sum 11.
Input: arr[] = [-2, -4]
Output: -2
Explanation: The subarray [-2] has the largest sum -2.
class Solution {
int maxSubarraySum(int[] arr) {
// Code here
int n=arr.length;
int currSum = arr[0];
int maxSoFar = arr[0];
for (int i = 1; i < n; i++) {
currSum = Math.max(arr[i], currSum + arr[i]);
maxSoFar = Math.max(maxSoFar, currSum);
}
return maxSoFar;
}
}