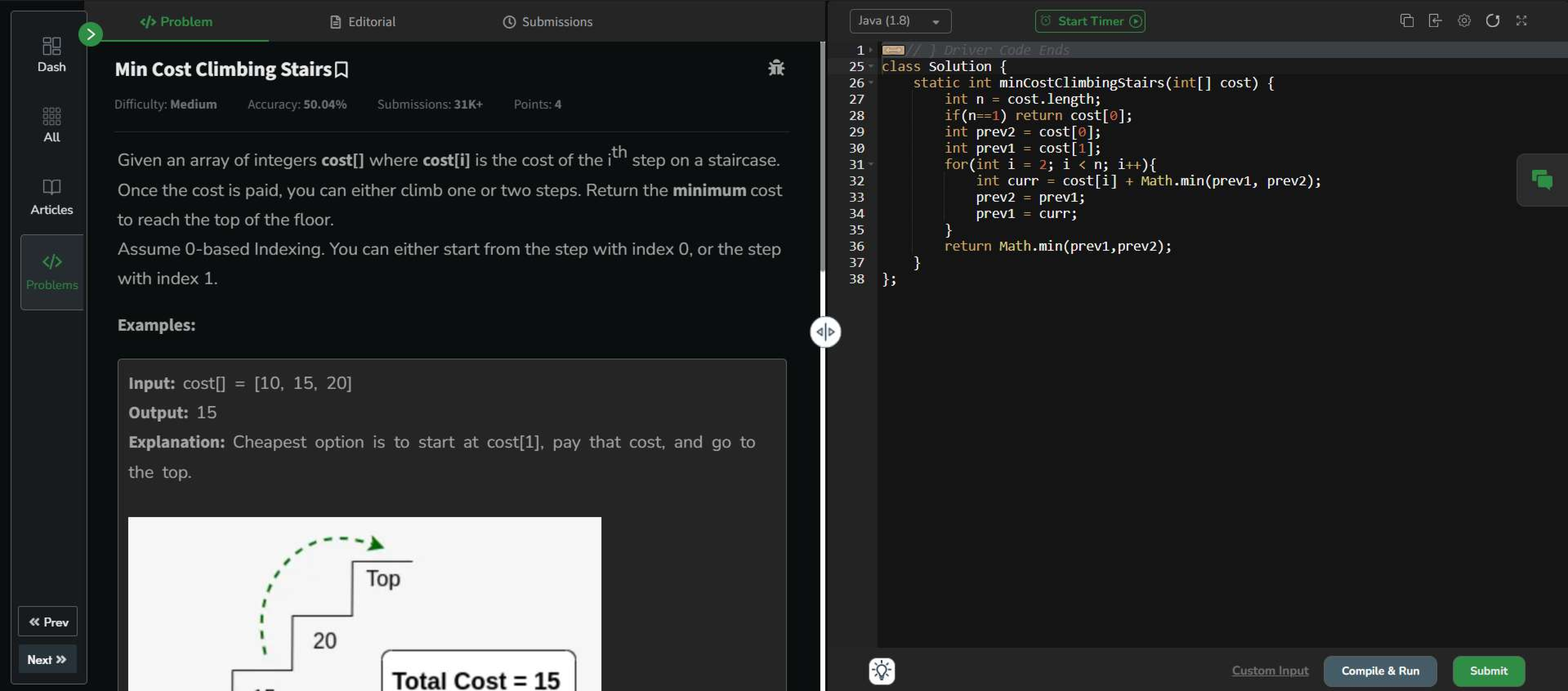

Min Cost Climbing Stairs

Difficulty: **Medium** Accuracy: **50.04%** Submissions: **31K+** Points: **4**

Given an array of integers **cost[]** where **cost[i]** is the cost of the ith step on a staircase. Once the cost is paid, you can either climb one or two steps. Return the **minimum** cost to reach the top of the floor.

Assume 0-based Indexing. You can either start from the step with index 0, or the step with index 1.

Examples:

Input: cost[] = [10, 15, 20]
Output: 15
Explanation: Cheapest option is to start at cost[1], pay that cost, and go to the top.

```java
// } Driver Code Ends
class Solution {
    static int minCostClimbingStairs(int[] cost) {
        int n = cost.length;
        if(n==1) return cost[0];
        int prev2 = cost[0];
        int prev1 = cost[1];
        for(int i = 2; i < n; i++){
            int curr = cost[i] + Math.min(prev1, prev2);
            prev2 = prev1;
            prev1 = curr;
        }
        return Math.min(prev1,prev2);
    }
};
```