

## AVAILABLE LESSONS:

*Lesson 1*  
Iterations

*Lesson 2*  
Arrays

*Lesson 3*  
Time Complexity

*Lesson 4*  
Counting Elements

*Lesson 5*  
Prefix Sums

*Lesson 6*  
Sorting

*Lesson 7*  
Stacks and Queues

*Lesson 8*  
Leader

*Lesson 9*  
Maximum slice problem

*Lesson 10*  
Prime and composite numbers

*Lesson 11*  
Sieve of Eratosthenes

AMBITIOUS

## MinAbsSum

START

Given array of integers, find the lowest absolute sum of elements.

Programming language: Python ▼

For a given array  $A$  of  $N$  integers and a sequence  $S$  of  $N$  integers from the set  $\{-1, 1\}$ , we define  $\text{val}(A, S)$  as follows:

$$\text{val}(A, S) = |\text{sum}\{A[i] * S[i] \text{ for } i = 0..N-1\}|$$

(Assume that the sum of zero elements equals zero.)

For a given array  $A$ , we are looking for such a sequence  $S$  that minimizes  $\text{val}(A, S)$ .

Write a function:

```
def solution(A)
```

that, given an array  $A$  of  $N$  integers, computes the minimum value of  $\text{val}(A, S)$  from all possible values of  $\text{val}(A, S)$  for all possible sequences  $S$  of  $N$  integers from the set  $\{-1, 1\}$ .

For example, given array:

```
A[0] = 1
A[1] = 5
A[2] = 2
A[3] = -2
```

your function should return 0, since for  $S = [-1, 1, -1, 1]$ ,  $\text{val}(A, S) = 0$ , which is the minimum possible value.

Assume that:

- $N$  is an integer within the range  $[0..20,000]$ ;
- each element of array  $A$  is an integer within the range  $[-100..100]$ .

Complexity:

- expected worst-case time complexity is  $O(N * \max(\text{abs}(A))^2)$ ;
- expected worst-case space complexity is  $O(N + \sum(\text{abs}(A)))$ , beyond input storage (not counting the storage required for input arguments).

Elements of input arrays can be modified.

Copyright 2009–2017 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

*Lesson 12*  
Euclidean algorithm

*Lesson 13*  
Fibonacci numbers

*Lesson 14*  
Binary search  
algorithm

*Lesson 15*  
Caterpillar method

*Lesson 16*  
Greedy algorithms

*Lesson 17*  
**Dynamic  
programming**

*Lesson 90*  
Tasks from Indeed  
Prime 2015  
challenge

*Lesson 91*  
Tasks from Indeed  
Prime 2016  
challenge

*Lesson 92*  
Tasks from Indeed  
Prime 2016 College  
Coders challenge

*Lesson 99*  
Future training

**For programmers**

Lessons Challenges  
Terms FAQ

**For companies**

Tour About us  
Pricing Jobs  
Blog Terms  
Privacy Cookies  
API

**Sign up for our newsletter:**

Information about upcoming challenges, solutions and lessons directly in your inbox.

Your email

Sign up

**Social:****Contact us:**

For customer support queries:  
UK +44 (0) 208 970 78 68  
US 1-415-466-8085  
support@codility.com

For sales queries:  
UK +44 (0) 208 970 78 67  
US 1-415-466-8085  
sales@codility.com

---

© 2009–2017 Codility Ltd., registered in England and Wales (No. 7048726). VAT ID GB981191408. Registered office:  
107 Cheapside, London EC2V 6DN