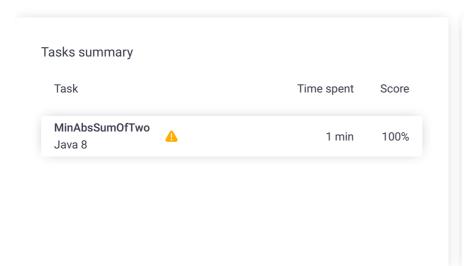
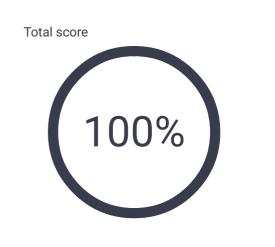
Codility_

CodeCheck Report: trainingCSSJQM-K75

Test Name:

Summary Timeline Check out Codility training tasks





Tasks Details

1. MinAbsSumOfTwo **Task Score** Correctness Performance Find the minimal 100% 100% 100% absolute value of a sum of two elements.

Task description

Let A be a non-empty array consisting of N integers.

The abs sum of two for a pair of indices (P, Q) is the absolute value |A[P] + A[Q]|, for $0 \le P \le Q < N$.

For example, the following array A:

A[0] = 1

A[1] = 4

A[2] = -3

has pairs of indices (0, 0), (0, 1), (0, 2), (1, 1), (1, 2), (2, 2). The abs sum of two for the pair (0, 0) is A[0] + A[0] = |1 + 1| = 2. The abs sum of two for the pair (0, 1) is A[0] + A[1] = |1 + 4| = 5. The abs sum of two for the pair (0, 2) is A[0] + A[2] = |1 + (-3)| =2.

The abs sum of two for the pair (1, 1) is A[1] + A[1] = |4 + 4| = 8. The abs sum of two for the pair (1, 2) is A[1] + A[2] = |4 + (-3)| =

The abs sum of two for the pair (2, 2) is A[2] + A[2] = |(-3) + (-3)|= 6.

Write a function:

Solution

Programming language used: Java 8 Total time used: 1 minutes Effective time used: 1 minutes Notes: not defined yet Task timeline ∇ 20:53:37 20:54:13 Code: 20:54:13 UTC, java, show code in pop-up final, score: 100 // you can also use imports, for example: import java.util.*;

```
class Solution { public int solution(int[]
A); }
```

that, given a non-empty array A consisting of N integers, returns the minimal abs sum of two for any pair of indices in this array.

For example, given the following array A:

```
A[0] = 1

A[1] = 4

A[2] = -3
```

the function should return 1, as explained above.

Given array A:

```
A[0] = -8
A[1] = 4
A[2] = 5
A[3] = -10
A[4] = 3
```

the function should return |(-8) + 5| = 3.

Write an efficient algorithm for the following assumptions:

- N is an integer within the range [1..100,000];
- each element of array A is an integer within the range [-1,000,000,000..1,000,000,000].

Copyright 2009–2023 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Test results - Codility

```
// you can write to stdout for debugging purpo
 5
     // System.out.println("this is a debug message
 6
 7
     class Solution {
 8
             public int solution(int[] A) {
9
                      final int N = A.length;
10
                      Arrays.sort(A);
11
12
13
                      int left = 0;
                      int right = N - 1;
14
15
16
                      int minAbsSum = Math.abs(A[let
17
                      while (left <= right) {
18
                              int sum = A[left] + A
19
                              minAbsSum = Math.min(r
20
                              if (sum <= 0) {
                                       left++;
21
22
                              } else {
23
                                       right--;
                              }
24
25
                      }
26
27
                      return minAbsSum;
28
             }
29
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity:

O(N * log(N))

olla	pse all	Example tests	
•	example1 first example	V	ок
1.	0.004 s OK		
•	example2 second example	V (OK
1.	0.004 s OK		
colla	pse all	Correctness tests	
•	extreme_single sequences of 1 eler		ок
1.	0.004 s OK		
2.	0.004 s OK		
3.	0.004 s OK		
•	extreme_double sequences of 2 eler		OK
1.	0.004 s OK		
2.	0.008 s OK		
3.	0.004 s OK		

Test results - Codility

•	positive_small only positive numbers	•	ОК		
1.	0.004 s OK				
•	negative_small only negative numbers	~	ОК		
1.	0.008 s OK				
collapse all Performance tests					
•	random_small random sequence, length = ~1000	~	ОК		
1.	0.008 s OK				
•	random_medium random sequence, length = ~10,000	~	ОК		
1.	0.052 s OK				
•	arithmetic_medium arithemtic sequence, length = ~10,000	V	ОК		
1.	0.124 s OK				
•	random_large random sequence, length = ~100,000	~	ОК		
1.	0.468 s OK				
•	extreme_large sequence of MAX_INT, length = ~100,000	~	ок		
1.	0.452 s OK				
•	arithmetic_large arithmetic sequence, length = ~100,000	~	ок		
1.	0.440 s OK				
•	constant_distance constant distance between all elements, length = 100,000	~	ОК		
1.	0.380 s OK				