Codility_

CodeCheck Report: trainingQ8ZGDH-GPC

Test Name:

Check out Codility training tasks

Summary Timeline

Tasks summary

Task	Time spent	Score
MinAbsSumOfTwo Java 8	50 min	100%

Total score

100%

Tasks Details

Medium

1. **MinAbsSumOfTwo**
Find the minimal absolute value of a sum of two elements.

Task Score	Correctness	Performance
100%	100%	100%

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 ≤ P ≤ 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)| = 1.
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:	50 minutes
Effective time used:	50 minutes
Notes:	*not defined yet*

Task timeline

18:38:14 19:27:46

Code: 19:27:45 UTC, java, final, score: **100** show code in pop-up

```
// 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].


```
3
4   // you can write to stdout for debugging purpo
5   // System.out.println("this is a debug message
6
7   class Solution {
8       /**
9            * Find the index of the most left ite
10           *
11           * @param array
12           * @param minValue
13           * @param left     The left index (inc
14           * @param right    The right index (ir
15           * @return The index of the most left
16           *         found.
17           */
18      private static int findMinIndex(int[] arra
19                  int minIndex = right + 1;
20                  while (left <= right) {
21                          int mid = (left + righ
22                          if (array[mid] >= min\
23                                  minIndex = mic
24                                  right = mid -
25                          } else {
26                                  left = mid + 1
27                          }
28                  }
29
30                  return minIndex;
31          }
32
33          public int solution(int[] A) {
34                  final int N = A.length;
35
36                  Arrays.sort(A);
37
38                  int minIndexNonNegative = finc
39                  if (minIndexNonNegative == 0)
40                          return 2 * A[0];
41                  } else if (minIndexNonNegative
42                          return -2 * A[N-1];
43                  }
44
45                  int i = minIndexNonNegative;
46                  int j = minIndexNonNegative -
47                  int minAbsSum = Math.min(2 * A
48                  if (minAbsSum == 0) {
49                          return 0;
50                  }
51
52                  for (; i < N; i++) {
53                          int localMinAbs = Inte
54                          int newLocalMinAbs;
55                          while (j >= 0 && (newl
56                                  localMinAbs =
57                                  j--;
58                          }
59                          if (localMinAbs < minA
60                                  minAbsSum = lo
61                                  if (minAbsSum
62                                          return
63                                  }
64                          }
65
66                          j++; // back one posit
67                  }
68
69
70                  return minAbsSum;
71          }
72  }
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: **O(N * log(N))**

collapse all **Example tests**

▼ example1 ✔ OK
first example

1. 0.004 s OK

▼ example2 ✔ OK
second example

1. 0.004 s OK

collapse all **Correctness tests**

▼ extreme_single ✔ OK
sequences of 1 elements

1. 0.008 s OK
2. 0.004 s OK
3. 0.004 s OK

▼ extreme_double ✔ OK
sequences of 2 elements

1. 0.008 s OK
2. 0.008 s OK
3. 0.004 s OK

▼ positive_small ✔ OK
only positive numbers

1. 0.008 s OK

▼ negative_small ✔ OK
only negative numbers

1. 0.004 s OK

collapse all **Performance tests**

▼ random_small ✔ OK
random sequence, length = ~1000

1. 0.008 s OK

▼ random_medium ✔ OK
random sequence, length = ~10,000

1. 0.052 s OK

▼ arithmetic_medium ✔ OK
arithemtic sequence, length = ~10,000

1. 0.120 s OK

▼ random_large ✔ OK
random sequence, length = ~100,000

1. 0.452 s OK

▼

extreme_large ✔ OK

sequence of MAX_INT, length = ~100,000

1. 0.436 s OK

▼ arithmetic_large ✔ OK

arithmetic sequence, length = ~100,000

1. 0.444 s OK

▼ constant_distance ✔ OK

constant distance between all elements, length = 100,000

1. 0.376 s OK