Open Feedback Dialog

codility

Training center

Check out Codility training tasks

Demo ticket

Session

ID: demoZSQ3V6-Z7E
Time limit: 120 min.

Status: closed

Created on: 2014-03-23 04:22 UTC
Started on: 2014-03-23 04:22 UTC
Finished on: 2014-03-23 04:29 UTC

Tasks in test

1 | {} ~~MinAbsSumOfTwo~~

Task score

100%

Test score

100%

100 out of 100 points

HARD

1. MinAbsSumOfTwo

Find the minimal absolute value of a sum of two elements.

score: 100 of 100

Task description

Let A be a non-empty zero-indexed 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:

```
def solution(A)
```

that, given a non-empty zero-indexed 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.

Solution

Programming language used: Python

Total time used: 7 minutes

Effective time used: 7 minutes

Notes: correct functionality and scalability

Task timeline

04:22:37 04:29:13

Code: 04:29:13 UTC, py, final, score: **100.00**

```
01. def solution(A):
02.     A.sort()
03.     N = len(A)
04.
05.     i = 0
06.     j = N-1
07.     min_abs = 1000000000 * 2
08.
09.     while i <= j:
10.         s = A[j] + A[i]
11.         min_abs = min(min_abs, abs(s))
12.         if s > 0:
13.             j -= 1
14.         elif s < 0:
15.             i += 1
16.         else:
17.             break
18.
19.     return min_abs
```

Analysis

Assume that:

- 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].

Complexity:

- expected worst-case time complexity is O(N*log(N));
- expected worst-case space complexity is O(1), beyond input storage (not counting the storage required for input arguments).

Elements of input arrays can be modified.

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

test	time	result
example1 first example	0.050 s.	**OK**
example2 second example	0.050 s.	**OK**
extreme_single sequences of 1 elements	0.050 s.	**OK**
extreme_double sequences of 2 elements	0.050 s.	**OK**
positive_small only positive numbers	0.050 s.	**OK**
negative_small only negative numbers	0.050 s.	**OK**
random_small random sequence, length = ~1000	0.050 s.	**OK**
random_medium random sequence, length = ~10,000	0.070 s.	**OK**
arithmetic_medium arithemtic sequence, length = ~10,000	0.130 s.	**OK**
random_large random sequence, length = ~100,000	0.280 s.	**OK**
extreme_large sequence of MAX_INT, length = ~100,000	0.280 s.	**OK**
arithmetic_large arithmetic sequence, length = ~100,000	0.340 s.	**OK**
constant_distance constant distance between all elements, length = 100,000	0.330 s.	**OK**

Training center