

### Congratulations

You have completed a Codility training test.

[Sign up for our newsletter!](#)
[Like us on Facebook!](#)

## Training ticket

### Session

ID: trainingK7TQQC-RFK  
Time limit: 120 min.

### Status: closed

Created on: 2016-06-04 19:53 UTC  
Started on: 2016-06-04 19:53 UTC  
Finished on: 2016-06-04 19:55 UTC

### Tasks in test

1 | **TapeEquilibrium**  
Submitted in: Java

Correctness

100%

Performance

100%

Task score

100%

100%

100 out of 100 points

EASY

### 1. TapeEquilibrium

Minimize the value  $|A[0] + \dots + A[P-1] - (A[P] + \dots + A[N-1])|$ .

score: 100 of 100



#### Task description

A non-empty zero-indexed array  $A$  consisting of  $N$  integers is given. Array  $A$  represents numbers on a tape.

Any integer  $P$ , such that  $0 < P < N$ , splits this tape into two non-empty parts:  $A[0], A[1], \dots, A[P-1]$  and  $A[P], A[P+1], \dots, A[N-1]$ .

The *difference* between the two parts is the value of:  $|A[0] + A[1] + \dots + A[P-1] - (A[P] + A[P+1] + \dots + A[N-1])|$

In other words, it is the absolute difference between the sum of the first part and the sum of the second part.

For example, consider array  $A$  such that:

```
A[0] = 3
A[1] = 1
A[2] = 2
A[3] = 4
A[4] = 3
```

We can split this tape in four places:

- $P = 1$ , difference =  $|3 - 10| = 7$
- $P = 2$ , difference =  $|4 - 9| = 5$
- $P = 3$ , difference =  $|6 - 7| = 1$
- $P = 4$ , difference =  $|10 - 3| = 7$

Write a function:

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

that, given a non-empty zero-indexed array  $A$  of  $N$  integers, returns the minimal difference that can be achieved.

For example, given:

```
A[0] = 3
```

#### Solution

Programming language used: Java

Total time used: 2 minutes

2

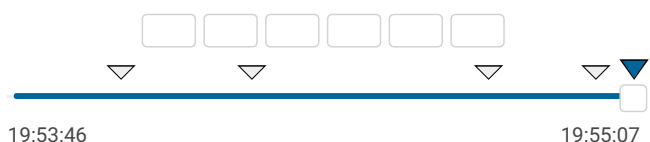
Effective time used: 2 minutes

2

Notes: *not defined yet*

#### Task timeline

2



Code: 19:55:07 UTC, java, final,  
score: 100

[show code in pop-up](#)

```
1 // you can also use imports, for example:
2 // import java.util.*;
3
4 // you can write to stdout for debugging purposes, e.g.
5 // System.out.println("this is a debug message");
6
7 import java.util.Arrays;
8
9 class Solution {
10
11     public int solution(int[] A) {
12         int diff = Integer.MAX_VALUE;
13         int nsu = 0;
14         int summ = Arrays.stream(A).sum();
15         int tmp = 0;
```

A[1] = 1  
A[2] = 2  
A[3] = 4  
A[4] = 3

the function should return 1, as explained above.

Assume that:

- N is an integer within the range [2..100,000];
- each element of array A is an integer within the range [-1,000..1,000].

Complexity:

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

Elements of input arrays can be modified.

Copyright 2009–2016 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

```
16     for (int i = 1; i<A.length; i++) {
17         nsu += A[i-1];
18         summ -= A[i-1];
19         tmp = Math.abs(summ - nsu);
20         if (tmp < diff) {
21             diff = tmp;
22         }
23     }
24 }
25 return diff;
26 }
27
28 }
```

Analysis summary

The solution obtained perfect score.

Analysis



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

Example tests	
▶ example	✓ OK
example test	
Correctness tests	
▶ double	✓ OK
two elements	
▶ simple_positive	✓ OK
simple test with positive numbers, length = 5	
▶ simple_negative	✓ OK
simple test with negative numbers, length = 5	
▶ small_random	✓ OK
random small, length = 100	
▶ small_range	✓ OK
range sequence, length = ~1,000	
▶ small	✓ OK
small elements	
Performance tests	
▶ medium_random1	✓ OK
random medium, numbers from 0 to 100, length = ~10,000	
▶ medium_random2	✓ OK
random medium, numbers from -1,000 to 50, length = ~10,000	
▶ large_ones	✓ OK
large sequence, numbers from -1 to 1, length = ~100,000	
▶ large_random	✓ OK
random large, length = ~100,000	
▶ large_sequence	✓ OK
large sequence, length = ~100,000	
▶ large_extreme	✓ OK
large test with maximal and minimal values, length = ~100,000	