

Candidate Report: trainingHDQ2V2-ATU

Check out Codility training tasks

Test Name:

Summary

Timeline

Tasks summary

Task	Time spent	Score
TapeEquilibrium C	46 min	61%

Total score

61%

Tasks Details

Easy	1. TapeEquilibrium Minimize the value $ A[0] + \dots + A[P-1] - (A[P] + \dots + A[N-1]) $.	Task Score 61%	Correctness 57%	Performance 66%
------	---	-------------------	--------------------	--------------------

Task description

A non-empty 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:

```
int solution(int A[], int N);
```

Solution

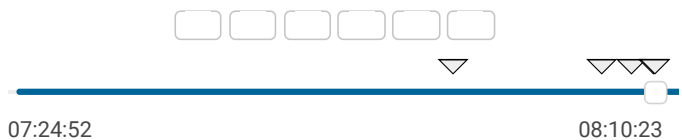
Programming language used: C

Total time used: 46 minutes ?

Effective time used: 46 minutes ?

Notes: not defined yet

Task timeline ?



Code: 08:10:23 UTC, c, final, [show code in pop-up](#)
score: 61

```
1 // you can write to stdout for debugging purposes, e.g.
2 // printf("this is a debug message\n");
3 #define the_abs(x) (x<0? -(x): x)
4
5 int solution(int A[], int N) {
6     // write your code in C99 (gcc 6.2.0)
7     int diff = 0, cnt = 0, tail_cnt = N-1;
8     for (int i=0; i<N; i++){
```

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

For example, given:

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

the function should return 1, as explained above.

Write an **efficient** algorithm for the following assumptions:

- 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].

Copyright 2009–2020 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Test results - Codility

```
9 //printf( "%d<%d\n", the_abs(diff+A[cnt]), the_abs
10
11 if ( the_abs(diff+A[cnt]) <= the_abs(diff-A[tail_c
12     diff += A[cnt++];
13 else
14     diff -= A[tail_cnt--];
15 }
16
17 return the_abs(diff);
18 }
```

Analysis summary

The following issues have been detected: wrong answers.

For example, for the input [-1000, 1000] the solution returned a wrong answer (got 0 expected 2000).

Analysis ?

expand all		Example tests
▶	example example test	✓ OK
expand all		Correctness tests
▼	double two elements	✗ WRONG ANSWER got 0 expected 2000

1.	0.001 s	WRONG ANSWER, got 0 expected 2000
2.	0.001 s	OK
3.	0.001 s	OK
▶	simple_positive simple test with positive numbers, length = 5	✓ OK
▶	simple_negative simple test with negative numbers, length = 5	✓ OK
▶	simple_boundary only one element on one of the sides	✓ OK
▼	small_random random small, length = 100	✗ WRONG ANSWER got 269 expected 39

1.	0.001 s	WRONG ANSWER, got 269 expected 39
▶	small_range range sequence, length = ~1,000	✓ OK
▼	small small elements	✗ WRONG ANSWER got 0 expected 20

1.	0.001 s	WRONG ANSWER, got 0 expected 20
expand all		Performance tests
▶	medium_random1 random medium, numbers from 0 to 100, length = ~10,000	✓ OK
▶	medium_random2 random medium, numbers from -1,000 to 50, length = ~10,000	✓ OK
▼	large_ones large sequence, numbers from -1 to 1, length = ~100,000	✗ WRONG ANSWER got 228 expected 0

1.	0.004 s	WRONG ANSWER, got 228 expected 0
2.	0.004 s	WRONG ANSWER, got 147 expected 1

Test results - Codility

▼ large_random		✗ WRONG ANSWER
random large, length = ~100,000		got 202635 expected 1
<hr/>		
1.	0.004 s	WRONG ANSWER, got 202635 expected 1
2.	0.004 s	WRONG ANSWER, got 34394 expected 2
<hr/>		
▶ large_sequence		✓ OK
large sequence, length = ~100,000		
<hr/>		
▶ large_extreme		✓ OK
large test with maximal and minimal values, length = ~100,000		

The PDF version of this report that may be downloaded on top of this site may contain sensitive data including personal information. For security purposes, we recommend you remove it from your system once reviewed.