Codility_

CodeCheck Report: trainingCJFCU4-TNW

Test Name:

Check out Codility training tasks

Summary Timeline AI Assistant Transcript

Tasks summary

Task	Time spent	Score
MinPerimeterRectangle C++	4 min	100%

Total score

Tasks Details

1. MinPerimeterRectangle

Easy

Find the minimal perimeter of any rectangle whose area equals N.

Task Score	Correctness	Performance
100%	100%	100%

Task description

An integer N is given, representing the area of some rectangle.

The *area* of a rectangle whose sides are of length A and B is A * B, and the *perimeter* is 2 * (A + B).

The goal is to find the minimal perimeter of any rectangle whose area equals N. The sides of this rectangle should be only integers.

For example, given integer N = 30, rectangles of area 30 are:

- (1, 30), with a perimeter of 62,
- (2, 15), with a perimeter of 34,
- (3, 10), with a perimeter of 26,
- (5, 6), with a perimeter of 22.

Write a function:

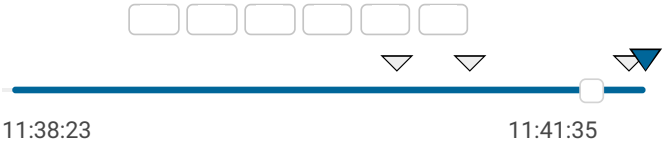
```
int solution(int N);
```

that, given an integer N, returns the minimal perimeter of any rectangle whose area is exactly equal to N.

Solution

Programming language used:	C++
Total time used:	4 minutes
Effective time used:	4 minutes
Notes:	*not defined yet*

Task timeline



11:38:23 11:41:35

Code: 11:41:35 UTC, cpp, final, score: **100** show code in pop-up

For example, given an integer N = 30, the function should return 22, as explained above.

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

- N is an integer within the range [1..1,000,000,000].


```
1  // you can use includes, for example:
2  // #include <algorithm>
3  #include <cmath>
4  #include <climits>
5  // you can write to stdout for debugging purp
6  // cout << "this is a debug message" << endl;
7
8  int solution(int N) {
9      // area: A*B
10     // perimeter: 2(A*B)
11     int minimum = INT_MAX;
12
13     for (int A = 1; A <= sqrt(N); ++A) {
14         if (N % A == 0) {
15             int B = N / A;
16             int perimeter = 2 * (A + B);
17             minimum = min(minimum, perimeter)
18         }
19     }
20
21     return minimum;
22 }
```

Analysis summary

The solution obtained perfect score.

Analysis

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

expand all	Example tests	
▶	example example test	✔ OK
expand all	**Correctness tests**	
▶	extreme_min N = 1 test	✔ OK
▶	simple1 N = 36 test	✔ OK
▶	simple2 N = 48 test	✔ OK
▶	simple3 N = 101 test	✔ OK
▶	small N = 1,234 test	✔ OK
expand all	**Performance tests**	
▶	medium N = 4,564,320 test	✔ OK
▶	prime1 N = 15,486,451 test	✔ OK
▶	square N = 100,000,000 test	✔ OK
▶	prime2 N = 982,451,653 test	✔ OK
▶	extreme_max N = 1,000,000,000 test	✔ OK