Codility_

CodeCheck Report: trainingXC54Z4-D3Y

Test Name:

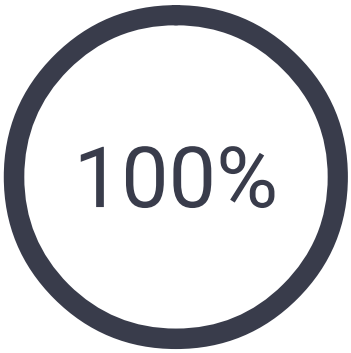
Check out Codility training tasks

Summary Timeline

Tasks summary

Task	Time spent	Score
BinaryGap Java 8 ⚠	66 min	100%

Total score



Tasks Details

Easy

1. BinaryGap

Find longest sequence of zeros in binary representation of an integer.

Task Score	Correctness	Performance
100%	100%	Not assessed

Task description

A *binary gap* within a positive integer N is any maximal sequence of consecutive zeros that is surrounded by ones at both ends in the binary representation of N.

For example, number 9 has binary representation `1001` and contains a binary gap of length 2. The number 529 has binary representation `1000010001` and contains two binary gaps: one of length 4 and one of length 3. The number 20 has binary representation `10100` and contains one binary gap of length 1. The number 15 has binary representation `1111` and has no binary gaps. The number 32 has binary representation `100000` and has no binary gaps.

Write a function:

```
class Solution { public int solution(int N); }
```

that, given a positive int
binary gap. The functior
binary gap.

For example, given N = 1
because N has binary re
longest binary gap is of
should return 0, becaus
and thus no binary gaps

Solution

Programming language used:	Java 8
Total time used:	66 minutes
Effective time used:	66 minutes
Notes:	*not defined yet*

Task timeline

13:57:35

show code in pop-up

for example:

How likely are you to recommend Codility to your friends and colleagues?

0 1 2 3 4 5 6 7 8 9 10

Not at all likely *Extremely likely*

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

- N is an integer within the range [1..2,147,483,647].


```
4  // you can write to stdout for debugging purposes,
5  // System.out.println("this is a debug message");
6
7  class Solution {
8      public int solution(int N) {
9          String binary = Integer.toBinaryString(N);
10         int count = 0;
11         int tmpCount = 0;
12         for (int i = 0; i < binary.length(); i++)
13             if (binary.charAt(i) == '0') {
14                 if (i > 0 && binary.charAt(i - 1)
15                     tmpCount++;
16                 } else {
17                     if (tmpCount > 0) tmpCount++;
18                 }
19             } else if (binary.charAt(i) == '1') {
20                 if (tmpCount > 0 && tmpCount > cou
21                     count = tmpCount;
22                 }
23                 tmpCount = 0;
24             }
25         }
26         return count;
27     }
28 }
29
30
31
```

Analysis summary

The solution obtained perfect score.

Analysis

expand all	Example tests	
▶	example1 example test n=1041=10000010001_2	✓ OK
▶	example2 example test n=15=1111_2	✓ OK
▶	example3 example test n=32=100000_2	✓ OK
expand all	Correctness tests	
▶	extremes n=1, n=5=101_2 and n=2147483647=2**31-1	✓ OK
▶	trailing_zeroes n=6=110_2 and n=328=101001000_2	✓ OK
▶	power_of_2 n=5=101_2, n=16=2**4 and n=1024=2**10	✓ OK
▶	simple1 n=9=1001_2 and n=11=1011_2	✓ OK
▶	simple2 n=19=10011 and n=42=101010_2	✓ OK
▶	simple3	✓ OK
		OK
		OK

	and n=9=1001_2	
▶	medium3 n=66561=10000010000000001_2	✓ OK
▶	large1 n=6291457=11000000000000000000001_2	✓ OK
▶	large2 n=74901729=10001110110111010001100001	✓ OK
▶	large3 n=805306373=11000000000000000000000000101_2	✓ OK
▶	large4 n=1376796946=10100100001000001000000100010010_2	✓ OK
▶	large5 n=1073741825=10000000000000000000000000001_2	✓ OK
▶	large6 n=1610612737=11000000000000000000000000001_2	✓ OK