

TASKS DETAILS

EASY	1. BinaryGap Find longest sequence of zeros in binary representation of an integer.	Task Score	Correctness	Performance
		60%	60%	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.

Write a function:

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

that, given a positive integer N, returns the length of its longest binary gap. The function should return 0 if N doesn't contain a binary gap.

For example, given N = 1041 the function should return 5, because N has binary representation 10000010001 and so its longest binary gap is of length 5.

Assume that:

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

Complexity:

- expected worst-case time complexity is $O(\log(N))$;
- expected worst-case space complexity is $O(1)$.

Solution

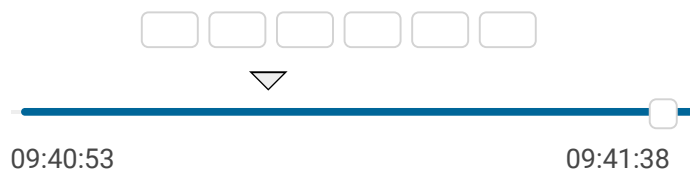
Programming language used: Java

Total time used: 1 minutes ?

Effective time used: 1 minutes ?

Notes: not defined yet

Task timeline



Code: 09:41:38 UTC, java, [show code in pop-up](#)
final, score: 60

```

1 // you can also use imports, for example:
2 import java.util.*;
3 import java.math.BigInteger;
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         BigInteger number = new BigInteger(Integer.toString(N));
10        BigInteger denominator = new BigInteger("2");
11
12        int binaryGap = 0;
13        int longestBinaryGap = 0;
14
15        while (number.intValue() != 1)
16            number = number.divide(denominator);

```

Test results - Codility

```
17         if (number.mod(denomin:
18             binaryGap++;
19         }
20
21         if (number.mod(denomin:
22             if (longestBin:
23                 longes:
24                 binary:
25             }
26         }
27     }
28     return longestBinaryGap;
29 }
30 }
```

Analysis summary

The following issues have been detected: wrong answers.

For example, for the input 16 the solution returned a wrong answer (got 3 expected 0).

Analysis ?

expand all	Example tests	
▶	example1 example test n=1041=10000010001_2	✓ OK
▶	example2 example test n=15=1111_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	✗ WRONG ANSWER got 3 expected 2
▶	power_of_2 n=5=101_2, n=16=2**4 and n=1024=2**10	✗ WRONG ANSWER got 3 expected 0
▶	simple1 n=9=1001_2 and n=11=1011_2	✓ OK
▶	simple2 n=19=10011 and n=42=101010_2	✓ OK
▶	simple3 n=1162=10010001010_2 and n=5=101_2	✓ OK
▶	medium1 n=51712=110010100000000_2 and n=20=10100_2	✗ WRONG ANSWER got 8 expected 2
▶		

medium2	<div><div></div><div>✖ WRONG ANSWER</div></div>
n=561892=10001001001011100100_2 and n=9=1001_2	got 5 expected 3
▶ medium3	<div><div></div><div>✔ OK</div></div>
n=66561=10000010000000001_2	
▶ large1	<div><div></div><div>✔ OK</div></div>
n=6291457=11000000000000000000001_2	
▶ large2	<div><div></div><div>✖ WRONG ANSWER</div></div>
n=74901729=10001110110111010001100001	got 5 expected 4
▶ large3	<div><div></div><div>✔ OK</div></div>
n=805306373=11000000000000000000000101_2	
▶ large4	<div><div></div><div>✖ WRONG ANSWER</div></div>
n=1376796946=10100100001000001000000100010010_2	got 9 expected 5
▶ large5	<div><div></div><div>✔ OK</div></div>
n=1073741825=10000000000000000000000000000001_2	
▶ large6	<div><div></div><div>✔ OK</div></div>
n=1610612737=11000000000000000000000000000001_2	