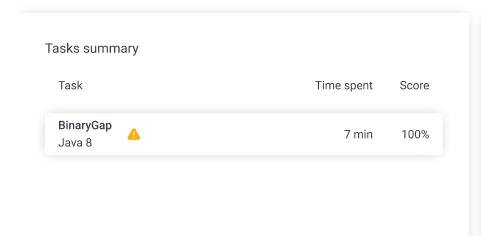
# Codility\_

## CodeCheck Report: training953E4R-2R2

Test Name:

Al Assistant Transcript Summary Timeline





Check out Codility training tasks

#### **Tasks Details**

### 1. BinaryGap

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

Task Score

100%

Correctness

Performance

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

#### Solution

Programming language used: Total time used: 7 minutes Effective time used: 7 minutes not defined yet Notes:

Task timeline



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. Given N = 32 the function should return 0, because N has binary representation '100000' and thus no binary gaps.

Write an efficient algorithm for the following assumptions:

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

Copyright 2009–2024 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

```
Code: 09:18:10 UTC, java,
                                     show code in pop-up
final, score: 100
1
     // you can also use imports, for example:
2
     // import java.util.*;
3
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 n = Integer.toBinaryString(N);
             int mi = 0;
10
11
             int ci = 0;
12
             for (int i=0;i<n.length();i++){</pre>
13
                 if(n.charAt(i)=='1'){
14
                     if(ci>mi){
15
                         mi=ci;
16
                     }
17
                     ci =0;
                 }
18
                 else
19
20
                     ci++;
21
             }
22
             return mi;
         }
23
24
```

#### Analysis summary

The solution obtained perfect score.

#### Analysis

expand all Example tests				
•	example1		✓ OK	
	example test n=104	1=10000010001_2		
•	example2		✓ OK	
	example test n=15=	1111_2		
•	example3		✓ OK	
	example test n=32=	100000_2		
expand all Correctness tests				
<b>•</b>	extremes		✓ OK	
	n=1, n=5=101_2 and	i		
	n=2147483647=2**	31-1		
•	trailing_zeroes		✓ OK	
	n=6=110_2 and n=3	28=101001000_2		
•	power_of_2		✓ OK	
	n=5=101_2, n=16=2	**4 and		
	n=1024=2**10			
•	simple1		✓ OK	
	n=9=1001_2 and n=	11=1011_2		
<b>•</b>	simple2		✓ OK	
	n=19=10011 and n=	42=101010_2		
<b>•</b>	simple3		✓ OK	
	n=1162=100100010	)10_2 and		

n=5=101_2	
► medium1 n=51712=110010100000000_2 and n=20=10100_2	√ OK
► medium2 n=561892=10001001001011100100_2 and n=9=1001_2	√ OK
► medium3 n=66561=1000001000000001_2	✓ OK
► large1 n=6291457=1100000000000000000000000000000000000	√ OK
► large2 n=74901729=10001110110111010001 1100001	√ OK
► large3 n=805306373=110000000000000000000000000000000000	√ OK
► large4 n=1376796946=101001000010000010 0000100010010_2	√ OK
► large5 n=1073741825=1000000000000000000000000000000000000	√ OK
► large6 n=1610612737=110000000000000000000000000000000000	√ OK