cødility

Candidate Report: Anonymous

Test Name:

SUMMARY TIMELINE

Test Score Tasks in Test

100 out of 100 points

100%

BinaryGap
Submitted in: C#

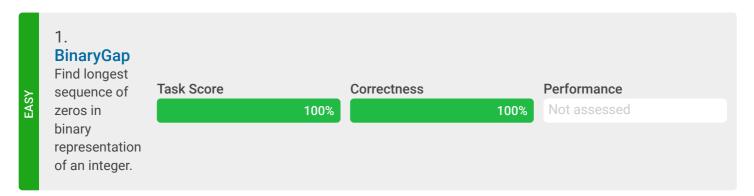
2 min

Time Spent

100%

Task Score

TASKS DETAILS



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:

Solution

Task timeline		?			
Notes:	not defined yet				
Effective time used:	2 minutes	0			
Total time used:	2 minutes	•			
Programming language used:	C#				

```
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. 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 2009–2019 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

21:55:58 21:57:32

Code: 21:57:32 UTC, cs, final, score: **100**

show code in pop-up

, 00010. 100

```
1
     using System;
 2
     // you can also use other imports, for example
 3
     // using System.Collections.Generic;
 4
     // you can write to stdout for debugging purpo
 5
 6
     // Console.WriteLine("this is a debug message"
 7
 8
     class Solution {
         public int solution(int N) {
 9
10
              if (N <= 0)
11
                               {
12
                                        throw new Argui
13
                               }
14
15
                               var digits = Convert.To
16
17
                               int biggestGap = 0;
18
                               int gap = 0;
19
                               var state = SearchState
20
                               foreach (var digit in
21
22
                                        if (digit == '
23
24
                                                switch
25
                                                {
26
27
28
29
30
31
32
33
34
35
                                                }
36
                                        }
37
                                        else
38
                                                switch
39
40
                                                {
41
42
43
44
45
46
47
48
49
50
51
52
53
                                                }
54
55
                                        }
56
                               }
57
58
                               return biggestGap;
59
         }
60
```

Analysis summary

The solution obtained perfect score.

Analysis ?

	Turnele Ace		
expa	and all Example tes		
•	example1 example test n=1041=10000010001_2	✓	OK
•	example2 example test n=15=1111_2	√	OK
•	example3 example test n=32=100000_2	✓	OK
ехра	and all Correctness to	est	S
•	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 n=1162=10010001010_2 and n=5=101_2	√	OK
•	medium1 n=51712=110010100000000_2 and n=20=10100_2	√	OK
•	medium2 n=561892=1000100100101110010 0_2 and n=9=1001_2	√	ОК
•	medium3 n=66561=10000010000000001_2	√	OK
>	large1 n=6291457=1100000000000000000000000000000000000	√	ОК

	00001_2	
•	large2 n=74901729=10001110110111010 0011100001	✓ OK
•	large3 n=805306373=11000000000000000 0000000000101_2	√ OK
>	large4 n=1376796946=101001000010000 0100000100010010_2	√ OK
•	large5 n=1073741825=100000000000000 00000000000000001_2	✓ OK
•	large6 n=1610612737=110000000000000 00000000000000001_2	✓ OK