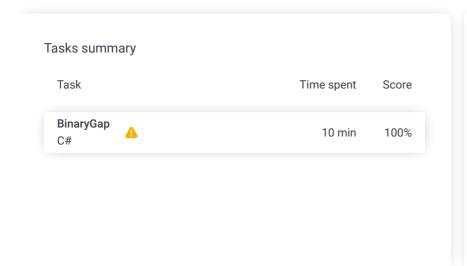
Codility_

CodeCheck Report: trainingRHV6CY-M2E

Test Name:

Check out Codility training tasks

Summary Timeline 💩 Al Assistant Transcript





Tasks Details

1. BinaryGap

Easy

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

Correctness

100%

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 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

Solution

Programming language used: C#

Total time used: 10 minutes

Effective time used: 10 minutes

Notes: not defined yet

Task timeline

10:56:16 11:06:06

Code: 11:06:05 UTC, cs, show code in pop-up final, score: 100

1 using System;
2 // you can also use other imports, for example:

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.

Test results - Codility

```
// using System.Collections.Generic;
5
    // you can write to stdout for debugging purposes,
6
    // Console.WriteLine("this is a debug message");
    class Solution {
         public int solution(int N) {
9
10
             // Implement your solution here
11
             int ans=0;
12
            int count=0;
             int startIndex=-1;
13
14
             while(N>0)
15
16
                 int lastbit= N & 1;
                 if(lastbit==1){
17
18
                     if(startIndex!=-1)
19
20
                          if(count-startIndex-1>ans)
21
                             ans=count-startIndex-1;
22
                     startIndex=count;
23
24
25
                 count++;
26
                 N=N/2;
             }
27
28
             return ans;
29
         }
30
     }
```

Analysis summary

The solution obtained perfect score.

Analysis

expa	nd all	Example tes	ls	
•	example1 example test n=104	1=10000010001_2		OK
•	example2 example test n=15=	1111_2	✓	OK
>	example3 example test n=32=	100000_2	✓	ОК
ехра	ind all	Correctness to	ests	3
>	extremes n=1, n=5=101_2 and n=2147483647=2**		✓	ОК
•	trailing_zeroes n=6=110_2 and n=3	28=101001000_2	✓	OK
•	power_of_2 n=5=101_2, n=16=2 n=1024=2**10	**4 and	✓	ОК
>	simple1 n=9=1001_2 and n=	11=1011_2	√	ОК
>	simple2 n=19=10011 and n=	-42=101010_2	✓	OK
>	simple3 n=1162=100100010 n=5=101_2	010_2 and	√	OK
>	medium1 n=51712=11001010 n=20=10100_2	00000000_2 and	✓	OK

Test results - Codility

rest results - obuility	
medium2	DK
medium3 n=66561=1000001000000001_2	✓ OK
► large1 n=6291457=1100000000000000000000000000000000000	√ OK
large2 n=74901729=1000111011011101000 11100001	√ OK
► large3 n=805306373=110000000000000000000000000000000000	√ OK
large4 n=1376796946=10100100001000001 00000100010010_2	√ OK
► large5 n=1073741825=1000000000000000000000000000000000000	√ OK
► large6 n=1610612737=110000000000000000000000000000000000	√ OK