codility

Training center

Congratulations

You have completed a Codility training test.

Sign up for our newsletter!

Like us on Facebook!

Training ticket

Session

ID: trainingTSZFVC-ZPQ
Time limit: 120 min.

Status: closed

Created on: 2017-03-18 20:17 UTC
Started on: 2017-03-18 20:17 UTC
Finished on: 2017-03-18 21:12 UTC

Style Assessment BETA

We'll email you a report with insights into this candidate's coding style within 5 working days.

Tasks in test	Correctness	Performance	Task score
1 BinaryGap Submitted in: Python	86%	not assessed	86%

Test score

86%

86 out of 100 points

EASY

1. BinaryGap

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

score: 86 of 100

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:

```
def solution(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.

Solution

Programming language used: Python

Total time used: 55 minutes

Effective time used: 55 minutes

Notes: *not defined yet*

Task timeline

20:17:49 21:12:15

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

Not at all likely *Extremely likely*

Complexity:

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


```
def solution(N):
        BinArray = []
        Array = []
        flag = 0
        counter = 0
        ## convert N to binary
        while not N==0:
                if N%2==0:
                        BinArray.append(0)
                else:
                        BinArray.append(1)
                N = N/2 ## update N

        ## reverse BinArray
        BinArray.reverse()

        ## counter for the gap
        for i in BinArray:
                if i==1:
                        if flag==0:
                                flag=1
                        else:
                                Array.append(counter)
                                counter=0
                else:
                        if flag==1:
                                counter=counter+1
        ## return
        return max(Array)
    # pass
```

Analysis summary

The following issues have been detected: runtime errors.

For example, for the input `1` the solution terminated unexpectedly.

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	✘ RUNTIME ERROR tested program terminated unexpectedly
▶	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	✘ RUNTIME ERROR tested program terminated unexpectedly
▶	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

large1 n=6291457=110000000000000000000001_2	✔ OK	
▶ large2 n=74901729=10001110110111010001110000 01	✔ OK	
▶ large3 n=805306373=11000000000000000000000000 000101_2	✔ OK	
▶ large4 n=1376796946=10100100001000001000001 00010010_2	✔ OK	
▶ large5 n=1073741825=10000000000000000000000 00000001_2	✔ OK	
▶ large6 n=1610612737=11000000000000000000000 00000001_2	✔ OK	

Training center

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

Not at all likely *Extremely likely*