g center

lity training tasks

Congratulations

You have completed a Codility training test.

Tweet this!

I scored 100% in #python on @Codility!

https://codility.com/demo/take-sample-test/binary_gap/

Sign up for our newsletter!

Like us on Facebook!

Training ticket

Session

ID: trainingYP75UY-VD8
Time limit: 120 min.

Status: closed

Created on: 2017-08-28 17:51 UTC
Started on: 2017-08-28 17:51 UTC
Finished on: 2017-08-28 19:44 UTC

Tasks in test	Correctness	Performance	Task score
1 BinaryGap Submitted in: Python	100%	not assessed	100%

Test score ?

100%

100 out of 100 points

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

Not at all likely *Extremely likely*

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.

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

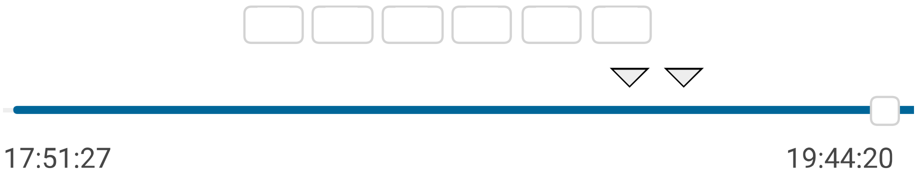
Programming language used: Python

Total time used: 113 minutes

Effective time used: 113 minutes

Notes: *not defined yet*

Task timeline



Code: 19:44:20 UTC, py, final, score: **100**

show code in pop-up

```
def solution(N):
  longest_binary_gap = 0
  current_binary_gap = 0
  one_remainder_reached = False

  while N > 0:
    #quotient, remainder = divmod(quotient, 2)
    quotient = N // 2
    remainder = N % 2
    N = quotient # must assign after get remainder

    # until we have remainder of > 0 dont have to do the rest
    if remainder == 1: one_remainder_reached = True

    if one_remainder_reached:

      if remainder == 0:
        current_binary_gap += 1
      else:
```

```
     # remainder is 1
     if (current_binary_gap > longest_binary_gap):
      longest_binary_gap = current_binary_gap
     current_binary_gap = 0

  return longest_binary_gap
```

Analysis summary

The solution obtained perfect score.

Analysis

expand all **Example tests**

▶ example1 ✓ OK

example test n=1041=10000010001_2

▶ example2 ✓ OK

example test n=15=1111_2

expand all **Correctness tests**

▶ extremes ✓ OK

n=1, n=5=101_2 and n=2147483647=2**31-1

▶ trailing_zeroes ✓ OK

n=6=110_2 and n=328=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

▶ simple2 ✓ OK

n=19=10011 and n=42=101010_2

▶ simple3 ✓ OK

n=1162=10010001010_2 and n=5=101_2

▶ medium1 ✓ OK

n=51712=110010100000000_2 and n=20=10100_2

▶

medium2 ✓ OK
n=561892=10001001001011100100_2 and
n=9=1001_2

▶ medium3 ✓ OK
n=66561=10000010000000001_2

▶ large1 ✓ OK
n=6291457=11000000000000000000001_2

▶ large2 ✓ OK
n=74901729=10001110110111010001110000
1

▶ large3 ✓ OK
n=805306373=1100000000000000000000000
00101_2

▶ large4 ✓ OK
n=1376796946=101001000010000010000010
0010010_2

▶ large5 ✓ OK
n=1073741825=100000000000000000000000
0000001_2

▶ large6 ✓ OK
n=1610612737=110000000000000000000000
0000001_2

Training center