


Tasks summary

| Task  | Time spent | Score |
|---|------------|-------|
| BinaryGap  | 34 min     | 100%  |

Total score

100%

Tasks Details

1. BinaryGap

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

Task Score

Correctness

Performance

100%

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:


```
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: 34 minutes 

Effective time used: 34 minutes 

Notes: not defined yet

Task timeline 

09:39:2310:12:56

Code: 10:12:56 UTC, cpp, final, score: 100 [show code in pop-up](#)

1

// you can use includes, for example:

2

// #include <algorithm>

https://app.codility.com/demo/results/trainingQ2VS9Y-A7R/

1/3

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

```
3
4 // you can write to stdout for debugging purposes
5 // cout << "this is a debug message" << endl;
6 int findMaxZeros(const std::string& S) {
7     int maxZeros = 0;
8     int count = 0;
9     bool counting = false;
10
11     for (char c : S) {
12         if (c == '1') {
13             if (counting) {
14                 maxZeros = max(maxZeros, count);
15                 count = 0;
16             } else {
17                 counting = true;
18             }
19         } else if (counting) {
20             count++;
21         }
22     }
23
24     return maxZeros;
25 }
26
27 int solution(int N) {
28     string binary = "";
29     for (int i = N; i > 0; i /= 2) {
30         int b = i % 2;
31
32         if (b == 1) {
33             binary = '1' + binary;
34         } else {
35             binary = '0' + binary;
36         }
37     }
38
39     return findMaxZeros(binary);
40 }
```

Analysis summary

The solution obtained perfect score.

Analysis

| Example tests                              |      |
|--|------|
| ▶ example1                                 | ✓ OK |
| example test n=1041=10000010001_2          |      |
| ▶ example2                                 | ✓ OK |
| example test n=15=1111_2                   |      |
| ▶ example3                                 | ✓ OK |
| example test n=32=100000_2                 |      |
| Correctness tests                          |      |
| ▶ extremes                                 | ✓ OK |
| n=1, n=5=101_2 and<br>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<br>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<br>n=5=101_2               |      |
| ▶ | medium1   | ✓ OK |
|   | n=51712=110010100000000_2 and<br>n=20=10100_2       |      |
| ▶ | medium2   | ✓ OK |
|   | n=561892=10001001001011100100_2<br>and n=9=1001_2   |      |
| ▶ | medium3   | ✓ OK |
|   | n=66561=10000010000000001_2                         |      |
| ▶ | large1  | ✓ OK |
|   | n=6291457=1100000000000000000<br>001_2              |      |
| ▶ | large2  | ✓ OK |
|   | n=74901729=1000111011011101000<br>11100001          |      |
| ▶ | large3  | ✓ OK |
|   | n=805306373=1100000000000000000<br>00000000101_2    |      |
| ▶ | large4  | ✓ OK |
|   | n=1376796946=10100100001000001<br>00000100010010_2  |      |
| ▶ | large5  | ✓ OK |
|   | n=1073741825=1000000000000000000<br>0000000000001_2 |      |
| ▶ | large6  | ✓ OK |
|   | n=1610612737=1100000000000000000<br>0000000000001_2 |      |