cødility

TASKS DETAILS

1.
BinaryGap

EASY

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:

```
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. Given N = 32 the function should return 0, because N has binary representation '100000' and thus no binary gaps.

Assume that:

- N is an integer within the range [1..2,147,483,647].

Complexity:

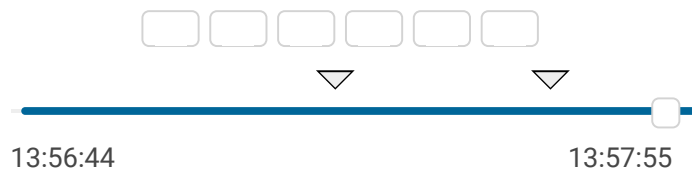- expected worst-case time complexity is O(log(N));

Solution

Programming language used:	Python
Total time used:	2 minutes
Effective time used:	2 minutes
Notes:	*not defined yet*

Task timeline

13:56:44   13:57:55

Code: 13:57:55 UTC, py, final, score: **100**   show code in pop-up

```python
 1
 2    def remove_borders(parts):
 3        first = parts[0]
 4        last = parts[-1]
 5        if not first or "0" in first:
 6            parts.pop(0)
 7        if not last or "0" in last:
 8            parts.pop()
 9        return parts
10
11
12    def solution(N):
13        binary_representation = f"{bin(N)}".re
14
15        binary_split = binary_representation.s
16        remove_borders(binary_split)
```

```
17
18        if binary_split:
19            binary_representation = max(binary
20            result = len(binary_representation
21        else:
22            result = 0
23
24        return result
```

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		
▶ example3		✔ OK
example test n=32=100000_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=1000100100101110010		
0_2 and n=9=1001_2		
▶ medium3		✔ OK
n=66561=10000010000000001_2		
▶ large1		✔ OK
n=6291457=110000000000000000		

- expected worst-case space complexity is O(1).

	00001_2	
▶	large2 n=74901729=10001110110111010 0011100001	✔ OK
▶	large3 n=805306373=1100000000000000 00000000000101_2	✔ OK
▶	large4 n=1376796946=101001000010000 0100000100010010_2	✔ OK
▶	large5 n=1073741825=100000000000000 0000000000000001_2	✔ OK
▶	large6 n=1610612737=110000000000000 0000000000000001_2	✔ OK