# Codility version: 1.1.20

00:01:20

nick: Glenn
Chen,

**English**

task 1 of 3

**next**

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 binary_gap(N)
```

that, given a positive integer N, returns the length of its longest binary gap. The function should
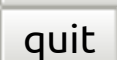
### Example test

```
example1 - OK
example2 - OK
```

**add test case**

```python
def binary_gap ( N ):
    '''O(n^2)'''

    #TODO: optimize bin(N)?
    #cached?
    bin_str = bin(N)
    bin_str = bin_str[2:]
    longest = 0
    curr = None
    current_gap = 0
    for i, char in enumerate(bin_str):
        if i == 0:
            prev = char #no need
            curr = char
            if curr == '0':
                current_gap += 1
                if current_gap > longest:
                    longest = current_gap
        else:
            curr = char
            if curr == '1':
                current_gap = 0
            if curr == '0':
                current_gap +=1
                if current_gap > longest:
```

Position:    Ln 1, Ch 1     Total:    Ln 28, Ch 686

- help
- verify
- submit task
- quit

Compilation successful