This code defines a class called "Solution" that has one public method named "solution". The method takes an integer input (N) and returns an integer output that represents the length of the longest binary gap in N.

Text

Description automatically generated

Here's a line-by-line breakdown of the code:

* class Solution {
  + This line starts the definition of a class named "Solution".
  + Everything within the curly braces that follow will be part of the class definition.



* public int solution(int N) {
  + This line starts the definition of a public method named "solution" that takes an integer input named "N".
  + The method returns an integer.



* String binaryRepresentation = Integer.toBinaryString(N);
  + This line converts the input integer N into a binary string and assigns it to a new variable called "binaryRepresentation".
  + The toBinaryString() method is a built-in Java method that converts an integer to a binary string representation.



* int longestBinaryGap = 0;
* int currentBinaryGap = 0;
  + These lines initialize two variables:
    - "longestBinaryGap" and "currentBinaryGap".
    - Both variables are integers and are set to 0.
  + These variables will be used to keep track of the longest binary gap found so far and the current binary gap being examined.

Text

Description automatically generated

* for (int i = 0; i < binaryRepresentation.length(); i++) {
  + This line starts a "for" loop that iterates through each character in the binary string representation of the input integer.
  + The loop starts at 0 and continues until i is less than the length of the binary representation string.



* char currentCharacter = binaryRepresentation.charAt(i);
  + This line gets the current character in the binary representation string at the current index "i" and assigns it to a new variable called "currentCharacter".
  + The charAt() method is a built-in Java method that gets the character at a specific index in a string.



* if (currentCharacter == '1') {

if (currentBinaryGap > longestBinaryGap) {

longestBinaryGap = currentBinaryGap;

}

currentBinaryGap = 0;

} else {

currentBinaryGap++;

}

* + These lines check whether the current character is a '1' or a '0'.
    - If it is a '1', the code checks whether the current binary gap is longer than the longest binary gap found so far.
    - If it is, the longest binary gap is updated.
  + The current binary gap is then reset to 0.
    - If the current character is a '0', the current binary gap is incremented by 1.

Text

Description automatically generated

* return longestBinaryGap;
  + This line returns the longest binary gap found so far.
  + It is the output of the "solution" method.



* }
  + This curly brace closes the "for" loop that iterated through the binary representation string.
* }
  + This curly brace closes the class definition.