Problem 868: Binary Gap

Problem Information

Difficulty: Easy
Acceptance Rate: 65.29%
Paid Only: No
Tags: Bit Manipulation

Problem Description

Given a positive integer `n`, find and return _the**longest distance** between any two **adjacent** _`1` _' s in the binary representation of _`n` _. If there are no two adjacent_`1` _' s, return _`0` _._

Two `1`'s are **adjacent** if there are only `0`'s separating them (possibly no `0`'s). The **distance** between two `1`'s is the absolute difference between their bit positions. For example, the two `1`'s in `"1001"` have a distance of 3.

Example 1:

Input: n = 22 **Output:** 2 **Explanation:** 22 in binary is "10110". The first adjacent pair of 1's is "_1_ 0 _1_ 10" with a distance of 2. The second adjacent pair of 1's is "10 _11_ 0" with a distance of 1. The answer is the largest of these two distances, which is 2. Note that "_1_ 01 _1_ 0" is not a valid pair since there is a 1 separating the two 1's underlined.

Example 2:

Input: n = 8 **Output:** 0 **Explanation:** 8 in binary is "1000". There are not any adjacent pairs of 1's in the binary representation of 8, so we return 0.

Example 3:

Input: n = 5 **Output:** 2 **Explanation:** 5 in binary is "101".

Constraints:

* `1 <= n <= 109`

Code Snippets

C++:

```
class Solution {
public:
int binaryGap(int n) {

}
};
```

Java:

```
class Solution {
public int binaryGap(int n) {

}
}
```

Python3:

```
class Solution:
def binaryGap(self, n: int) -> int:
```