

Power of two 2

To determine if a given integer (n) is a power of two without using loops or recursion, you can use bit manipulation.

Solution Explanation

An integer (n) is a power of two if and only if it is greater than zero and its binary representation has exactly one bit set. This can be checked using the following condition:

$[n > 0 \text{ \texttt{\&\&}} (n \& (n - 1)) == 0]$

Implementation in Python

Here's how you can implement it:

```
def is_power_of_two(n):  
    return n > 0 and (n & (n - 1)) == 0
```

Examples

1. Input: 1
1. Output: True
1. Explanation: ($2^0 = 1$)
2. Input: 16
2. Output: True
2. Explanation: ($2^4 = 16$)
3. Input: 3
3. Output: False
3. Explanation: 3 is not a power of two.

Constraints

Make sure to handle cases based on the constraints provided ($(-2^{31} \leq n \leq 2^{31} - 1)$). The above function satisfies these constraints directly since negative numbers will return False