Problem 2220: Minimum Bit Flips to Convert Number

Problem Information

Difficulty: Easy
Acceptance Rate: 87.71%
Paid Only: No
Tags: Bit Manipulation

Problem Description

A **bit flip** of a number `x` is choosing a bit in the binary representation of `x` and **flipping** it from either `0` to `1` or `1` to `0`.

* For example, for `x = 7`, the binary representation is `111` and we may choose any bit (including any leading zeros not shown) and flip it. We can flip the first bit from the right to get `110`, flip the second bit from the right to get `101`, flip the fifth bit from the right (a leading zero) to get `10111`, etc.

Given two integers `start` and `goal`, return _the**minimum** number of **bit flips** to convert _`start` _to_`goal`.

Example 1:

Input: start = 10, goal = 7 **Output:** 3 **Explanation:** The binary representation of 10 and 7 are 1010 and 0111 respectively. We can convert 10 to 7 in 3 steps: - Flip the first bit from the right: 101 _0_ -> 101 _1_. - Flip the third bit from the right: 1 _0_ 11 -> 1 _1_ 11. - Flip the fourth bit from the right: _1_ 111 -> _0_ 111. It can be shown we cannot convert 10 to 7 in less than 3 steps. Hence, we return 3.

Example 2:

Input: start = 3, goal = 4 **Output:** 3 **Explanation:** The binary representation of 3 and 4 are 011 and 100 respectively. We can convert 3 to 4 in 3 steps: - Flip the first bit from the right: 01 _1_ -> 01 _0_. - Flip the second bit from the right: 0 _1_ 0 -> 0 _0_ 0. - Flip the third bit from the right: _0_ 00 -> _1_ 00. It can be shown we cannot convert 3 to 4 in less than 3

steps. Hence, we return 3.

Constraints:

* `0 <= start, goal <= 109`

Note: This question is the same as [461: Hamming Distance.](https://leetcode.com/problems/hamming-distance/description/)

Code Snippets

C++:

```
class Solution {
public:
int minBitFlips(int start, int goal) {

}
};
```

Java:

```
class Solution {
public int minBitFlips(int start, int goal) {

}
}
```

Python3:

```
class Solution:
def minBitFlips(self, start: int, goal: int) -> int:
```