

0067. 二进制求和

👤 ITCharge ⌚ 大约 1 分钟

- 标签：位运算、数学、字符串、模拟
- 难度：简单

题目链接

- [0067. 二进制求和 - 力扣](#)

题目大意

给定两个二进制数的字符串 a 、 b 。计算 a 和 b 的和，返回结果也用二进制表示。

解题思路

这道题可以直接将 a 、 b 转换为十进制，相加后再转换为二进制数。

也可以利用位运算的一些知识，直接求和。

因为 a 、 b 为二进制的字符串，先将其转换为二进制数。

本题用到的位运算知识：

- 异或运算 $x \oplus y$ ：可以获得 $x + y$ 无进位的加法结果。
- 与运算 $x \& y$ ：对应位置为 1，说明 x 、 y 该位置上原来都为 1，则需要进位。
- 左移运算 $x \ll 1$ ：将 a 对应二进制数左移 1 位。

这样，通过 $x \oplus y$ 运算，我们可以得到相加后无进位结果，再根据 $(x \& y) \ll 1$ ，计算进位后结果。

进行 $x \oplus y$ 和 $(x \& y) \ll 1$ 操作之后判断进位是否为 0，若不为 0，则继续上一步操作，直到进位为 0。

最后将其结果转为 2 进制返回。

代码

py

```
class Solution:
    def addBinary(self, a: str, b: str) -> str:
        x = int(a, 2)
        y = int(b, 2)
        ans = 0
        while y:
            carry = ((x & y) << 1)
            x ^= y
            y = carry
        return bin(x)[2:]
```