Given two Binary Strings, a and b,

return their sum as a binary string

e.g.

a = "11", b = "1"

$$\begin{array}{r} \textcircled{1} \\ 11 \\ +\quad 1 \\ \hline 100 \end{array}$$

addition in base 2

$$\begin{array}{r} \textcircled{1} \\ 18 \\ +\quad 7 \\ \hline 5 \end{array}$$

e.g.

$$\begin{array}{r} \textcircled{1}\textcircled{1} \\ 1010 \\ +\ 1011 \\ \hline 10101 \end{array}$$

→ "11"

11

"01"

```
result = "" , carry = 0
first check lengths
if lengths are not equal:
        diffLengths = a.len - b.len
                if diffLength > 0:
                        "0" * diffLengths + b
                else:
                        "0" * (-diffLength) + a
```

```
for (let i=a.length-1; i>=0; i--) {
    currA = a[i]; // last elements in both strings
    currB = b[i];
    addition= int(currA) + int(currB) + carry
    if addition == 0||addition == 1 ;
        add addition to beginning of result string
    else
        add 0 to ————————
        carry = 1
}
```

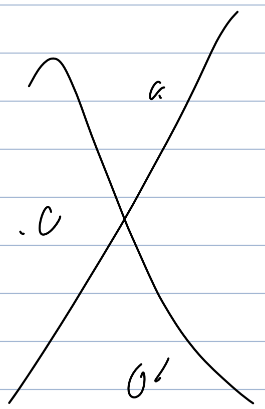
return result

O(1) space

O(N) ti

carry = ~~0~~ 1

```
 1 1 1
 0 0 1
 ─────
   0 0
```



does not work

neetcode

```
 1 1             1 1
   11              11
 +  1          +   11
 ----          ------
  100             110
```

reverse the strings a, b

```
for in range(max(a.length, b.length)):
    digitA = int(a[i]) if i < len(a) else 0
    digitB =   "          "        "
    total = digitA + digitB + carry
    char = str(total % 2)
    res = char + res
    carry = total // 2
if carry:
    result = '1' + res
return result
```