

累加数

累加数 是一个字符串，组成它的数字可以形成累加序列。

一个有效的 **累加序列** 必须 **至少** 包含 3 个数。除了最开始的两个数以外，字符串中的其他数都等于它之前两个数相加的和。

给你一个只包含数字 '0'-'9' 的字符串，编写一个算法来判断给定输入是否是 **累加数**。如果是，返回 true；否则，返回 false。

说明：累加序列里的数 **不会** 以 0 开头，所以不会出现 1, 2, 03 或者 1, 02, 3 的情况。

示例 1：

输入: "112358"

输出: true

解释: 累加序列为: 1, 1, 2, 3, 5, 8 。 $1 + 1 = 2$, $1 + 2 = 3$, $2 + 3 = 5$, $3 + 5 = 8$

示例 2：

输入: "199100199"

输出: true

解释: 累加序列为: 1, 99, 100, 199。 $1 + 99 = 100$, $99 + 100 = 199$

```
class Solution:
    def isAdditiveNumber(self, num: str) -> bool:
        # 循环查找 匹配
        for i in range(1, len(num)-1):
            for j in range(i+1, len(num)):
                if self.check_sum(num, i, j) == num:
                    return True
        return False

    def check_sum(self, num, i, j):
        # 合法性检测 首位数字不能为 0
        if len(num[:i]) >= 2 and num[0] == '0':
            return ""
        if len(num[i:j]) >= 2 and num[i] == '0':
            return ""
```

```
a=int(num[:i])# 1 i=1 取 0 位  
b=int(num[i:j])# 99 i=1 j=3 取 1 和 2 位  
ans=num[:j] #199 j=3 取 0 , 1, 2 位  
while len(ans)<len(num):  
    #a=b #a=99 100  
    #b=a+b #b=100 199
```

a,b=b,a+b #先计算=号的右边 b 的值，a+b 的值，算好了，然后再分别赋值给 a 和 b 就可以了

```
ans+=str(b) #ans=199100199  
return ans
```