Given a string S of digits, such as S = "123456579", we can split it into a *Fibonacci-like sequence* [123, 456, 579].

Formally, a Fibonacci-like sequence is a list F of non-negative integers such that:

* 0 <= F[i] <= 2^31 - 1, (that is, each integer fits a 32-bit signed integer type);
* F.length >= 3;
* and F[i] + F[i+1] = F[i+2] for all 0 <= i < F.length - 2.

Also, note that when splitting the string into pieces, each piece must not have extra leading zeroes, except if the piece is the number 0 itself.

Return any Fibonacci-like sequence split from S, or return [] if it cannot be done.

**Example 1:**

**Input:** "123456579"

**Output:** [123,456,579]

**Example 2:**

**Input:** "11235813"

**Output:** [1,1,2,3,5,8,13]

**Example 3:**

**Input:** "112358130"

**Output:** []

**Explanation:** The task is impossible.

**Example 4:**

**Input:** "0123"

**Output:** []

**Explanation:** Leading zeroes are not allowed, so "01", "2", "3" is not valid.

**Example 5:**

**Input:** "1101111"

**Output:** [110, 1, 111]

**Explanation:** The output [11, 0, 11, 11] would also be accepted.

**Note:**

1. 1 <= S.length <= 200
2. S contains only digits.

给定一个数字字符串 S，比如 S = "123456579"，我们可以将它分成斐波那契式的序列 [123, 456, 579]。

形式上，斐波那契式序列是一个非负整数列表 F，且满足：

* 0 <= F[i] <= 2^31 - 1，（也就是说，每个整数都符合 32 位有符号整数类型）；
* F.length >= 3；
* 对于所有的0 <= i < F.length - 2，都有 F[i] + F[i+1] = F[i+2] 成立。

另外，请注意，将字符串拆分成小块时，每个块的数字一定不要以零开头，除非这个块是数字 0 本身。

返回从 S 拆分出来的所有斐波那契式的序列块，如果不能拆分则返回 []。

**示例 1：**

**输入：**"123456579"

**输出：**[123,456,579]

**示例 2：**

**输入:** "11235813"

**输出:** [1,1,2,3,5,8,13]

**示例 3：**

**输入:** "112358130"

**输出:** []

**解释:** 这项任务无法完成。

**示例 4：**

**输入：**"0123"

**输出：**[]

**解释：**每个块的数字不能以零开头，因此 "01"，"2"，"3" 不是有效答案。

**示例 5：**

**输入:** "1101111"

**输出:** [110, 1, 111]

**解释:** 输出 [11,0,11,11] 也同样被接受。

**提示：**

1. 1 <= S.length <= 200
2. 字符串 S 中只含有数字。