Problem 842: Split Array into Fibonacci Sequence

Problem Information

Difficulty: Medium
Acceptance Rate: 40.03%
Paid Only: No
Tags: String, Backtracking

Problem Description

You are given a string of digits `num`, such as `"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] < 231`, (that is, each integer fits in 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`.

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 `num`, or return `[]` if it cannot be done.

Example 1:

Input: num = "1101111" **Output:** [11,0,11,11] **Explanation:** The output [110, 1, 111] would also be accepted.

Example 2:

Input: num = "112358130" **Output:** [] **Explanation:** The task is impossible.

Example 3:

Input: num = "0123" **Output:** [] **Explanation:** Leading zeroes are not allowed, so "01", "2", "3" is not valid.

Constraints:

* `1 <= num.length <= 200` * `num` contains only digits.

Code Snippets

C++:

```
class Solution {
public:
vector<int> splitIntoFibonacci(string num) {

}
};
```

Java:

```
class Solution {
public List<Integer> splitIntoFibonacci(String num) {

}
}
```

Python3:

```
class Solution:
def splitIntoFibonacci(self, num: str) -> List[int]:
```