**Additive Number**

Additive number is a string whose digits can form additive sequence.

A valid additive sequence should contain **at least** three numbers. Except for the first two numbers, each subsequent number in the sequence must be the sum of the preceding two.

Given a string containing only digits '0'-'9', write a function to determine if it's an additive number.

class Solution **{**

public boolean isAdditiveNumber**(**String num**)** **{**

**return** isAdditiveNumber**(**num**,**0**,new** ArrayList**<>());**

**}**

private boolean isAdditiveNumber**(**String num**,** int curInx**,** ArrayList**<**String**>** numList**){**

**if(**curInx **==** num**.**length**()&&**numList**.**size**()** **>=** 3 **)** //Base Case

**return** **true;**

**for(**int i **=** curInx**;**i **<=** num**.**length**()-**1**;** i**++){** //Backtrack with all options

**if(**num**.**charAt**(**curInx**)==**'0' **&&** i**!=**curInx**)**

**break;**

String currNum **=** num**.**substring**(**curInx**,**i**+**1**);** //Get new Number

**if(**numList**.**size**()<=**1 **||** currNum**.**equals**(**

String**.**valueOf**(**

Long**.**parseLong**(**numList**.**get**(**numList**.**size**()-**1**))+**Long**.**parseLong**(**numList**.**get**(**numList**.**size**()-**2**))**

**))){**

numList**.**add**(**currNum**);**

**if(**isAdditiveNumber**(**num**,**i**+**1**,**numList**))** //Make the recursion until you get answer

**return** **true;**

numList**.**remove**(**numList**.**size**()-**1**);**

**}**

**}**

**return** **false;**//Return false if there no possible answer

**}**

**}**