## 2. Prime String

Given a string of length *n* consisting of digits [0-9], count the number of ways the given string can be split into prime numbers. The digits must remain in the order given and the entire string must be used. Each number must be in the range *2 to* $10^6$ inclusive, and may not contain leading zeros. Since the answer can be large, return the answer modulo $(10^9 + 7)$.

*Note: The initial string does not contain leading zeros.*

**Example**

*s = "11373"*

This string can be split into primes *6 different ways: [11, 37, 3], [113, 7, 3], [11, 3, 73], [11, 37, 3], [113, 73] and [11, 373].*

**Function Description**

Complete the function *countPrimeStrings* in the editor below.

countPrimeStrings has the following parameter(s):
  *string s:* a string of digits
**Return**
  *int:* the number of ways the string can be split into primes, modulo *1000000007, (*$10^9$*+7)*

**Constraints**

- $1 \le length\ of\ s \le 10^5$

▶ **Input Format For Custom Testin**

▼ **Sample Case 0**

**Sample Input For Custom Testing**

```
STDIN       Function
-----       --------
3175     → s = "3175"
```

**Sample Output**

```
3
```

**Explanation**

The *3* ways to split this string into prime numbers are *[31, 7, 5], [3, 17, 5], [317, 5]. 3 modulo (*$10^9$ *+ 7) = 3*

▶ **Sample Case 1**

Python 3                ● Autocomplete Ready ⓘ

```
 1 > #!/bin/python3 ...
10   #
11   # Complete the 'countPrimeStrings' function below.
12   #
13   # The function is expected to return an INTEGER.
14   # The function accepts STRING s as parameter.
15   #
16
17   def countPrimeStrings(s):
18       # Write your code here
19 > if __name__ == '__main__': ...
```

Line: 10 Col: 1

**Test Results**          **Custom Input**                    Run ▲    **Submit Code**