Problem 972: Equal Rational Numbers

Problem Information

Difficulty: Hard
Acceptance Rate: 45.56%
Paid Only: No
Tags: Math, String

Problem Description

Given two strings `s` and `t`, each of which represents a non-negative rational number, return `true` if and only if they represent the same number. The strings may use parentheses to denote the repeating part of the rational number.

A **rational number** can be represented using up to three parts: `<IntegerPart>`, `<NonRepeatingPart>`, and a `<RepeatingPart>`. The number will be represented in one of the following three ways:

* `<IntegerPart>` * For example, `12`, `0`, and `123`. * `<IntegerPart>**<.>**<NonRepeatingPart>` * For example, `0.5`, `1.`, `2.12`, and `123.0001`. * `<IntegerPart>**<.>**<NonRepeatingPart>**<(>**<RepeatingPart>**<)>**` * For example, `0.1(6)`, `1.(9)`, `123.00(1212)`.

The repeating portion of a decimal expansion is conventionally denoted within a pair of round brackets. For example:

* `1/6 = 0.16666666... = 0.1(6) = 0.1666(6) = 0.166(66)`.

Example 1:

Input: s = "0.(52)", t = "0.5(25)" **Output:** true **Explanation:** Because "0.(52)" represents 0.52525252..., and "0.5(25)" represents 0.52525252525..... , the strings represent the same number.

Example 2:

Input: s = "0.1666(6)", t = "0.166(66)" **Output:** true

Example 3:

Input: s = "0.9(9)", t = "1." **Output:** true **Explanation:** "0.9(9)" represents 0.999999999... repeated forever, which equals 1. [[See this link for an explanation.](https://en.wikipedia.org/wiki/0.999...)] "1." represents the number 1, which is formed correctly: (IntegerPart) = "1" and (NonRepeatingPart) = "".

Constraints:

* Each part consists only of digits. * The `<IntegerPart>` does not have leading zeros (except for the zero itself). * `1 <= <IntegerPart>.length <= 4` * `0 <= <NonRepeatingPart>.length <= 4` * `1 <= <RepeatingPart>.length <= 4`

Code Snippets

C++:

```
class Solution {
public:
bool isRationalEqual(string s, string t) {

}
};
```

Java:

```
class Solution {
public boolean isRationalEqual(String s, String t) {

}
}
```

Python3:

```
class Solution:
def isRationalEqual(self, s: str, t: str) -> bool:
```