166. Fraction to Recurring Decimal

March 10, 2016 | 37.3K views

Average Rating: 4 (51 votes)

Given two integers representing the numerator and denominator of a fraction, return the fraction in string format.

If the fractional part is repeating, enclose the repeating part in parentheses.

Example 1:

```
Input: numerator = 1, denominator = 2
Output: "0.5"
```

Example 2:

```
Input: numerator = 2, denominator = 1
Output: "2"
```

Example 3:

```
Input: numerator = 2, denominator = 3
Output: "0.(6)"
```

Summary

This is a straight forward coding problem but with a fair amount of details to get right.

Hints

1. No scary math, just apply elementary math knowledge. Still remember how to perform a *long division*?
2. Try a long division on $\frac{4}{9}$, the repeating part is obvious. Now try $\frac{4}{333}$. Do you see a pattern?
3. Be wary of edge cases! List out as many test cases as you can think of and test your code thoroughly.

Solution

Approach 1: Long Division

Intuition

The key insight here is to notice that once the remainder starts repeating, so does the divided result.

$$
\begin{array}{r}
0.16 \\
6\overline{)1.00} \\
\underline{0} \\
1\,0 \\
\underline{6} \\
40 \\
\underline{36} \\
4
\end{array}
\quad
\begin{array}{l}
\\
\\
\\
\Leftarrow remainder = 1, \text{mark 1 as seen at } position = 0. \\
\\
\Leftarrow remainder = 4, \text{mark 4 as seen at } position = 1. \\
\\
\Leftarrow remainder = 4 \text{ was seen before at } position = 1, \\
\text{so the fractional part starts repeating at } position = 1 \Rightarrow 1(6).
\end{array}
$$

Algorithm

You will need a hash table that maps from the remainder to its position of the fractional part. Once you found a repeating remainder, you may enclose the reoccurring fractional part with parentheses by consulting the position from the table.

The remainder could be zero while doing the division. That means there is no repeating fractional part and you should stop right away.

Just like the question Divide Two Integers, be wary of edge cases such as negative fractions and nasty extreme case such as $\frac{-2147483648}{-1}$.

Here are some good test cases:

Test case	Explanation
$\frac{0}{1}$	Numerator is zero.
$\frac{1}{0}$	Divisor is 0, should handle it by throwing an exception but here we ignore for simplicity sake.
$\frac{20}{4}$	Answer is a whole integer, should not contain the fractional part.
$\frac{1}{2}$	Answer is 0.5, no recurring decimal.
$\frac{-1}{4}$ or $\frac{1}{-4}$	One of the numerator or denominator is negative, fraction is negative.
$\frac{-1}{-4}$	Both numerator and denominator are negative, should result in positive fraction.
$\frac{-2147483648}{-1}$	Beware of overflow if you cast to positive.

Java

Copy

```java
public String fractionToDecimal(int numerator, int denominator) {
    if (numerator == 0) {
        return "0";
    }
    StringBuilder fraction = new StringBuilder();
    // If either one is negative (not both)
    if (numerator < 0 ^ denominator < 0) {
        fraction.append("-");
    }
    // Convert to Long or else abs(-2147483648) overflows
    long dividend = Math.abs(Long.valueOf(numerator));
    long divisor = Math.abs(Long.valueOf(denominator));
    fraction.append(String.valueOf(dividend / divisor));
    long remainder = dividend % divisor;
    if (remainder == 0) {
        return fraction.toString();
    }
    fraction.append(".");
    Map<Long, Integer> map = new HashMap<>();
    while (remainder != 0) {
        if (map.containsKey(remainder)) {
            fraction.insert(map.get(remainder), "(");
            fraction.append(")");
            break;
        }
        map.put(remainder, fraction.length());
        remainder *= 10;
```

Rate this article:

Previous Next

Comments: 19 Sort By

Type comment here... (Markdown is supported)

Preview Post

ChangRutgers ★21 December 10, 2017 6:52 AM

what's the space complexity?

21 Share Reply

SHOW 3 REPLIES

jabberwok ★4 September 7, 2017 6:19 PM

@Geminiiiiii

I can't read the original article because my subscription ran out, so I can't explain in details what is wrong with it. the only thing I remember that it used HashMap which was a huge overkill performance-wise. My idea was that all fractions are either finite like 0.123 or cyclic like 0.123333...(3) . Cyclic fractions

Read More

6 Share Reply

amaroraiko ★64 October 1, 2019 12:53 AM

The unit tests around integer min and max values are brutal.

4 Share Reply

SHOW 1 REPLY

ueabu ★3 November 14, 2019 11:50 AM

probably a dumb question but why do you multiply the remainder by 10 each time? I revised long division and I can't see anywhere that happens.

3 Share Reply

SHOW 1 REPLY

calvinchankf ★2925 June 20, 2019 9:25 AM

Here is my approach which beats 90.96% in python

math + hashtable

- do long division

Read More

1 Share Reply

oyugioho ★1 September 6, 2017 12:04 PM

I think he is referring to the technique of loop detection that we all know how to do for the question "Find the node where the loop starts for a singly linked list". Basically, try to get reminders two at a time for one and one at a time for another. When they are equal, we detect a loop, etc.

1 Share Reply

SHOW 1 REPLY

HT_Wang ★487 July 24, 2019 8:11 AM

Awesome coding question

1 Share Reply

magusliuj ★4 February 15, 2019 4:00 AM

```
// for test case 1/333
// frac 0.0
// map {1=2} coutains the position of where the remainder appears
```

Read More

0 Share Reply

sona123 ★43 January 17, 2019 10:03 AM

What is the time complexity ?

0 Share Reply

SHOW 1 REPLY

2016prep ★7 May 28, 2020 6:18 AM

37/39 test cases passed during mock session, only missed doing the math with longs

0 Share Reply

1 2