772. Basic Calculator III

Hard 615 229 Add to List Share

Implement a basic calculator to evaluate a simple expression string.

The expression string contains only non-negative integers, `'+'`, `'-'`, `'*'`, `'/'` operators, and open `'('` and closing parentheses `')'`. The integer division should **truncate toward zero**.

You may assume that the given expression is always valid. All intermediate results will be in the range of $[-2^{31}, 2^{31} - 1]$.

Note: You are not allowed to use any built-in function which evaluates strings as mathematical expressions, such as `eval()`.

Example 1:

```
Input: s = "1+1"
Output: 2
```

Example 2:

```
Input: s = "6-4/2"
Output: 4
```

Example 3:

```
Input: s = "2*(5+5*2)/3+(6/2+8)"
Output: 21
```

Example 4:

```
Input: s = "(2+6*3+5-(3*14/7+2)*5)+3"
Output: -12
```

Example 5:

```
Input: s = "0"
Output: 0
```

Constraints:

- $1 <= s <= 10^4$
- `s` consists of digits, `'+'`, `'-'`, `'*'`, `'/'`, `'('`, and `')'`.
- `s` is a **valid** expression.

Accepted 58,943 Submissions 131,775

Seen this question in a real interview before? Yes No

Companies

0 ~ 6 months 6 months ~ 1 year 1 year ~ 2 years

Amazon | 4 Facebook | 3 Uber | 2 ByteDance | 2

Related Topics

Similar Questions

```python
class Solution:
    # Function to find precedence
    # of operators.
    def precedence(self, op):
        if op == '+' or op == '-':
            return 1
        if op == '*' or op == '/':
            return 2
        return 0

    # Function to perform arithmetic
    # operations.
    def applyOp(self, a, b, op):
        if op == '+': return a + b
        if op == '-': return a - b
        if op == '*': return a * b
        if op == '/': return int(a / b)

    # Function that returns
```

Your previous code was restored fro

Test... Run Code ... Deb...

Watch Expressions

Local Variables

```
i: 0
ops: []
tokens: …
values: …
```

stdout