Lab Assignment – 9

Use the following information to code your assignment 9

Class TreeNode

Represents a node in the expression tree.

Attributes:

- `value (int | str)`: Stores either an operand (integer) or an operator (`+, -, *, /`).
- `left (TreeNode | None)`: Left child node.
- `right (TreeNode | None)`: Right child node.

Methods:

- `__init__(self, value: int | str)`:
  Initializes a TreeNode with a given value.

Class ExpressionTree

Builds and evaluates an expression tree from a postfix expression.

Attributes:

- `root (TreeNode | None)`: The root node of the constructed expression tree.

Methods:

`__init__(self, postfix_expr: str) -> None`

Initializes the expression tree by constructing it from the given postfix expression.

`construct_expression_tree(self, postfix: str) -> TreeNode`

Constructs an expression tree from a postfix expression. This method should be called in the constructor of the Expression tree to initialize the root.

Parameters:

- `postfix (str)`: A space-separated string of operands and operators in postfix notation.

Returns:

- `TreeNode | None`: The root of the constructed tree.

`evaluate_expression(self) -> int | float`

Returns:

- `int | float`: The evaluated result of the expression.
- Use recursion
- **Note: MAKE SURE YOU RETURN float('inf') WHENEVER YOU ENCOUNTER *DIVISION BY ZERO* (THERE ARE TESTS FOR CHECKING THIS).**

Returns the evaluated result of the expression tree. Use root which is available with self using self.root for evaluating

EXAMPLES:

EXPRESSION: "3 4 + 5 * 6 –"
OUTPUT: 29

EXPRESSION: "10 2 8 * + 3 -"
OUTPUT: 23

EXPRESSION: "7 3 1 + * 5 /"
OUTPUT: 5.6

EXPRESSION: "9 4 2 + * 8 / 3 -"
OUTPUT: 3.75

EXPRESSION: "5 1 2 + 4 * + 3 -"
OUTPUT: "14"

EXPRESSION: "6 2 / 3 4 * +"
OUTPUT: 15

`get_inorder_expression(self) -> str`

Returns:

- `str`: The infix expression with proper parentheses (uses self.root for getting the expression)

Example:

Example 1:

The tree is constructed using the string “3 4 + 5 * 6 –“
The output for this method is “(((3+4)*5)-6)”

Example 2:

The tree is constructed using the string “2 3 + 4 * 5 -“
The output for this method is “(((2+3)*4)-5)”