

Documentation

The Solution class provides a calculation method to evaluate a mathematical expression given as a string. This method adheres to specific constraints, ensuring that it can handle basic arithmetic operations such as addition and subtraction and nested expressions represented by parentheses. The input string can contain digits, operators (+, -), parentheses, and whitespace while guaranteeing that the expression is valid and well-formed.

Upon invocation, the calculate method initializes a stack, a variable for the current number being processed, a variable for the accumulated result, and a variable to track the last operation encountered. The method iterates through each character in the input string, performing actions based on the type of character detected. If a digit is encountered, it constructs the current number by taking into account any multi-digit numbers. When an operator or a parenthesis is identified, the method applies the last operation to the current number and updates the accumulated result accordingly.

For handling parentheses, the method employs the stack to save the current result and the last operation whenever it encounters an opening parenthesis. This allows the method to reset the result for evaluating the new expression enclosed within the parentheses. Upon reaching a closing parenthesis, it concludes the evaluation of the current expression, applying any remaining operations and restoring the previous result from the stack.

Throughout the execution, any whitespace characters are ignored to streamline the parsing process. The method concludes its evaluation by returning the final computed result, ensuring it fits within the constraints of a signed 32-bit integer. This implementation effectively mimics a basic calculator, leveraging a stack-based approach to manage nested expressions and perform arithmetic operations without relying on built-in evaluation functions. The overall structure allows for efficient processing of expressions with lengths up to 300,000 characters, making it robust and capable of handling a wide range of valid mathematical expressions.