Multiply Strings Documentation

Problem Statement:

Given two non-negative integers `num1` and `num2` represented as strings, the task is to return the product of `num1` and `num2`, also represented as a string.

Example:

- **Example 1:**
  - Input: `num1 = "2"`, `num2 = "3"`
  - Output: `"6"`

- **Example 2:**
  - Input: `num1 = "123"`, `num2 = "456"`
  - Output: `"56088"`

Constraints:

- 1 <= `num1.length`, `num2.length` <= 200
- `num1` and `num2` consist of digits only.
- Both `num1` and `num2` do not contain any leading zero, except the number 0 itself.

Approach:

The solution employs a straightforward approach where each digit of the first number (`num1`) is multiplied with each digit of the second number (`num2`). The result is then calculated by adding the products together with proper positioning considering their place values.

Algorithm:

1. Check if either `num1` or `num2` is "0". If so, return "0" since the product will be "0".
2. Initialize variables `len1` and `len2` to store the lengths of `num1` and `num2` respectively.
3. Initialize a result array of size `(len1 + len2)` with zeros.
4. Iterate over each digit of `num1` from right to left (i.e., from least significant digit to most significant digit).
    a. For each digit of `num1`, iterate over each digit of `num2` from right to left.
    b. Calculate the product of the current digits and add it to the corresponding positions in the result array.
5. Convert the result array to a string by removing leading zeros.
6. Return the final result string.

Implementation:

The solution is implemented using a class named `Solution` with a method named `multiply`. This method takes two string arguments `num1` and `num2` and returns their product as a string.

Complexity Analysis:

- Time Complexity: O(m * n), where m and n are the lengths of `num1` and `num2` respectively.
- Space Complexity: O(m + n) for storing the result array.