Code Challenge

Conduct “FizzBuzz” on a tree while traversing through it. Change the values of each of the nodes dependent on the current node’s value

Specifications

- Read all of these instructions carefully. Name things exactly as described.
- Do all your work in a public repository called `data-structures-and-algorithms`, with a well-formatted, detailed top-level README.md.
- Create a new branch in your repo called `fizzbuzz_tree`.
- Your top-level readme should contain a “Table of Contents” navigation to all of your challenges and implementations so far. (Don’t forget to update it!)
- This assignment should be completed within the `challenges` subdirectory of the repository.
- On your branch, create…
    - *C#*: a new .NET Core console project named `FizzBuzzTree`. Within your `Program.cs` create a new static method outside of `Main()` following the naming conventions below. Call your newly created method in `Main()` once complete.
    - *JavaScript*: a folder named `fizzBuzzTree` which contains a file called `fizz-buzz-tree.js`
    - *Python*: a folder named `fizz_buzz_tree` which contains a file called `fizz_buzz_tree.py`
    - *Java*: a folder named `FizzBuzzTree` which contains a file called `FizzBuzzTree.java`

- Include any language-specific configuration files required for this challenge to become an individual component, module, library, etc.
    - *NOTE: You can find an example of this configuration for your course in your class lecture repository.*

Feature Tasks

- Write a function called `FizzBuzzTree` which takes a tree as an argument.
- Without utilizing any of the built-in methods available to your language, determine weather or not the value of each node is divisible by 3, 5 or both, and change the value of each of the nodes:
    - If the value is divisible by 3, replace the value with “Fizz”
    - If the value is divisible by 5, replace the value with “Buzz”

    - If the value is divisible by 3 and 5, replace the value with “FizzBuzz”

- Return the tree with its new values.

- For explicitly-typed languages: Ensure your node values are of type `Object`, to hold either strings or integers.

Requirements

Ensure your complete solution follows the standard requirements.

1. Write unit tests
2. Follow the template for a well-formatted README
3. Submit the assignment following these instructions