# Code Challenge

Breadth-first Traversal.

## 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 `breadth_first`.
- 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#*: Extend your `BinaryTree` class according to the feature tasks below
    - *JavaScript*: Extend your `BinaryTree` class according to the feature tasks below
    - *Python*: Extend your `BinaryTree` class according to the feature tasks below
    - *Java*: Extend your `BinaryTree` class according to the feature tasks below

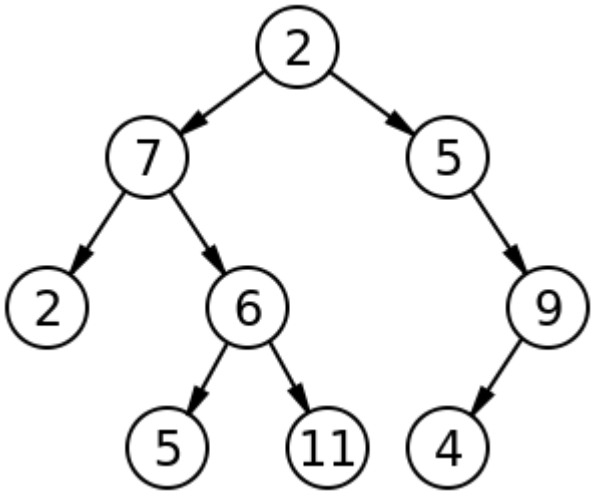
- Include any language-specific configuration files required for this challenge to become an individual component, module, library, etc.
    - *NOTE: You can find an example of this configuration for your course in your class lecture repository.*

## Feature Tasks

- Write a breadth first traversal method which takes a Binary Tree as its unique input. Without utilizing any of the built-in methods available to your language, traverse the input tree using a Breadth-first approach; print every visited node's value.

## Example

### Input




### Output

```
2
7
5
2
6
9
5
11
4
```

## 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