**Trees: Solving Problems with Trees**

To succeed at this challenge, you'll need to demonstrate that you can do the following:

* solve various problems with trees.

**Instructions**

Your goal for this checkpoint is to get the tests to pass.

To do so, you will be modifying the existing BinarySearchTree class to add methods that traverse the BST using in-order, pre-order, and post-order traversals.

**Existing files**

| **File path** | **Description** |
| --- | --- |
| src/BinarySearchTree.js | Contains the definition of the BinarySearchTree class. The constructor() method and methods from the previous checkpoint assignments have already been completed for you. |

**Tasks**

Complete the following tasks to pass the tests and this assignment.

In src/BinarySearchTree.js:

1. **Count Leaves**: Write a method to count the number of leaves in the tree. Your method should return a number as an output. If the tree only has a single node, return 1. Add your solution to the countLeaves() method.
2. **Balanced BST**: Write an algorithm that checks if a BST is height-balanced (i.e., the height of the left subtree differs from the height of the right subtree by no more than 1), returning the height of the BST if it is balanced, or -1 otherwise. Add your solution to the isBalancedBST() method.

Once these tasks are complete, all tests should pass.