**Testing report**

**Tree Traversal Testing:**

To test the three traversal algorithms, trees with different structures were created. These tests included: a large unbalanced tree, a tree with three children on a single node and path tree where each element contains a single child.

The large unbalanced tree tested that these traversals performed as expected when a tree is not balanced. The tree with three children was designed to ensure that the traversals still returned values in the correct order. Here we were testing especially to ensure left children were always visited first. A single path tree was also used.

As well as this, our testing included testing empty trees, as well as testing edge cases in a tree with only 2 nodes.

**Tree Properties:**

Testing was conducted using a variety of trees. This included a simple large tree with multiple nodes, an unbalanced tree and an incomplete tree.

The large tree was used to test methods and ensure that they still worked on a larger scale. This was especially targeted at counting methods such as finding height, leaf nodes and testing if the tree is a heap. Unbalanced tree tested that relevant functions could recognise the tree was not balanced. Tests were also included for boundary cases such as single nodes, empty trees and path trees.

**Self-balancing binary search tree:**

To test BST cases were written to test a path tree with 2 nodes, a tree with a single node and an empty tree. These test rare cases to ensure that the add and remove functions perform correctly in rare cases. A large binary search tree was also created to test that the add/remove function could handle inserting/removing values into a large tree.

A path tree was also created to test that the functions wouldn’t run if called on an unbalanced tree. Add and remove were called together to ensure there were no errors.

**Arithmetic Expressions:**

Cases were written to test each method to cover all possible bases. Ie. Returning negative results, division by zero. A large tree and a tree holding integer values were also tested to ensure that expected output was given. Tests were conducted to make sure the output was printed correctly for each function, that it is an arithmetic expression and can be evaluated.

|  |  |  |
| --- | --- | --- |
| **Test** | **Actual Output** | **Expected Output** |
| **Traversal Tests** |  |  |
| Large, Unbalanced tree tested with characters “a”-“h”. Tested with preOrder | "a", "b", "d", "f", "g", "h", "e", "c" | "a", "b", "d", "f", "g", "h", "e", "c" |
| Large, Unbalanced tree tested with characters ” a-h”. Tested with postOrder | "g", "h", "f", "d", "e", "b", "c", "a" | "g", "h", "f", "d", "e", "b", "c", "a" |
| Large, Unbalanced tree tested with characters “a-h”. Tested with inOrder | UnsupportedOperation  Exception | UnsupportedOperation  Exception |
| A non- binary tree tested with preOrder | "a", "b", "c", "d" | "a", "b", "c", "d" |
| A non- binary tree tested with postOrder | "c", "d", "e", "b", "a" | "c", "d", "e", "b", "a" |
| A non- binary tree tested with inOrder | UnsupportedOperation  Exception | UnsupportedOperation  Exception |
|  |  |  |
| **Properties Tests** |  |  |
| Height tested with a large tree | 3 | 3 |
| Height with depth constraint tested on incomplete tree |  |  |
| IscompleteBinary tested with a large, incomplete tree | false | false |
| Unbalanced tree tested with isHeap where min is true | true | true |
| Large balanced tree tested with isBalancedBinary | true | true |
|  |  |  |
| **BST Tests** |  |  |
| Test adding new numbers into a large balanced binary tree. | True and value added | True and value added |
| Test adding to an empty tree | True and value added | True and value added |
| Test removing from an empty tree |  | False and nothing removed |
| Test add and remove together. | Remove returns true and add returns true. | Remove returns true and add returns true. |
| Ensure tree is a balanced binary after removing and adding |  | true |
|  |  |  |
| **Arithmetic Expressions** |  |  |
| Test Division by zero | Positive infinity | Positive infinity |
| Testing expressions without outermost brackets | True and cases excepted | True |
| Testing a tree with integer values (invalid). Test isArithmetic | false | false |
| Testing with negative numbers  test 1 + 1.2 = 1.2 | -2.2 | -2.2 |
| Testing subtraction with negative result  test 5-(1+1) = 6 | -1.0 | -1.0 |
| Testing multiplication 3\*(1+1) = 6 | 6.0 | 6.0 |