Find the Minimum and Maximum Height of a Binary Search Tree

In the last challenge we described a scenario in which a tree could become unbalanced. To understand the concept of balance, let's take a look at another tree property: height. Height in a tree represents the distance from the root node to any given leaf node. Different paths in a highly branched tree structure may have different heights, but for a given tree there will be a minimum and maximum height. If the tree is balanced, these values will differ at most by one. This means that in a balanced tree, all the leaf nodes exist within the same level, or if they are not within the same level they are at most one level apart.

The property of balance is important for trees because it is what determines the efficiency of tree operations. As we explained in the last challenge, we face worst case time complexity for heavily unbalanced trees. Self-balancing trees are commonly used to account for this issue in trees with dynamic data sets. Common examples of these include AVL trees, red-black trees, and B-trees. These trees all contain additional internal logic which re-balance the tree when insertions or deletions create a state of imbalance.

Note: A similar property to height is depth, which refers to how far a given node is from the root node.

Write two methods for our binary tree: `findMinHeight` and `findMaxHeight`. These methods should return an integer value for the minimum and maximum height within a given binary tree, respectively. If the node is empty, let’s assign it a height of `-1` (that’s the base case). Finally, add a third method `isBalanced` which returns or depending on whether the tree is balanced or not. You can use the first two methods you just wrote to determine this.

🧪	The `BinarySearchTree` data structure should exist.
🧪	The binary search tree should have a method called `findMinHeight`.

⚗	The binary search tree should have a method called `findMaxHeight`.
⚗	The binary search tree should have a method called `isBalanced`.
⚗	The `findMinHeight` method should return the minimum height of the tree.
⚗	The `findMaxHeight` method should return the maximum height of the tree.
⚗	An empty tree should return a height of `-1`.
⚗	The `isBalanced` method should return `false` if the tree is an unbalanced binary search tree.
⚗	The `isBalanced` method should return `true` if the tree is a balanced binary search tree.