W4 - Data Structure & Algorithms

## Problem 1: Compute the height of a binary tree.

**Overview**: Detect the height of a binary tree.

**Definition of important concepts**

1. Binary tree: is defined as a tree data structure where each node has at most 2 children.
2. The height of a node in a binary tree is the largest number of edges in a path from a leaf node to a target node.
3. The height of a binary tree: is equal to the largest number of edges from the root to the most distant leaf node, or the height of the most distant node.

**Algorithm Steps**:

1. **Using recursion**
    - Since the tree is the binary tree, using recursion, we just check the height of the left subtree and the right subtree.
    - Take the maximum of the two numbers, and that is the height of the tree.
2. **Code**

```
class TreeNode {
   int value;
   TreeNode left, right;

   TreeNode(int value) {
       this.value = value;
       left = right = null;
   }
}

public class BinaryTree {
   TreeNode root;

   // Method to calculate the height of a binary tree
   int height(TreeNode node) {
       if (node == null)
           return 0;
       else {
           // Compute the height of each subtree
           int leftHeight = height(node.left);
           int rightHeight = height(node.right);

           // Use the larger one
           return Math.max(leftHeight, rightHeight) + 1;
       }
   }

}
```

## Problem 2: Implement Binary Search Tree (BST)

*↳ 1 cell hidden*

## Problem 3: Graph Representation and Traversal

*↳ 1 cell hidden*