2. Trees :: Binary Trees :: Introduction and Definitions

A **binary tree** is an ordered tree where each node has 0, 1 or 2 child nodes. The first child (if it exists) is called the **left child** and the second child (if it exists) is called the **right child**. A right child may exist without a left child and a left child may exist without a right child.

The subtree rooted at the left child of a node N is called the **left subtree** of N and the subtree rooted at the right child of N is called the **right subtree** of N.

A binary tree of height h is **balanced** if all leaf nodes are either at level h or level h - 1.

2. Trees :: Binary Trees :: Introduction and Definitions (continued)

A **full binary tree** is one in which each node is either a leaf (has 0 children) or has 2 children (in the case of interior nodes).

Recursively defined, a full binary tree is one in which subtree is a full binary tree:

```
Method isFull(In: Node pRoot) Returns Boolean
  If pRoot is a leaf Then Return true
  If pRoot has a left child Then leftFull ← isFull(pRoot.leftChild)
  Else leftFull ← false
  If pRoot has a right child Then rightFull ← isFull(pRoot.rightChild)
  Else rightFull ← false
  Return leftFull AND rightFull
End Method isFull
```

2. Trees :: Binary Trees :: Introduction and Definitions (continued)

A binary tree of height h is a **complete binary tree** if all levels are completely full, except possibly the last which if not full, has all of its nodes to the left side.