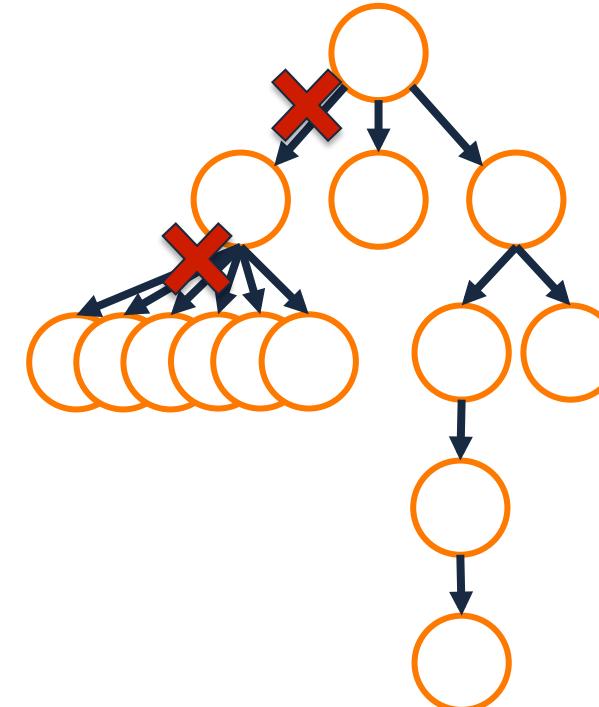
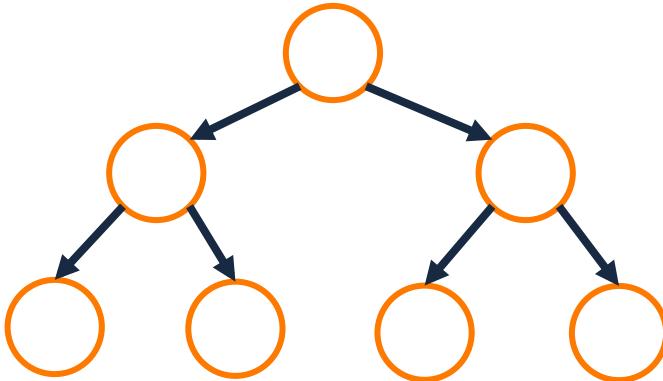
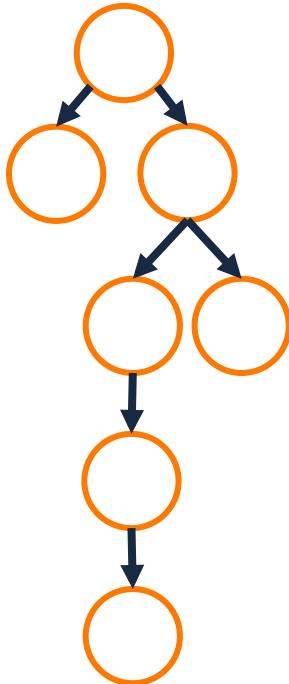


Binary Trees

Prof. Wade Fagen-Ulmschneider

I ILLINOIS

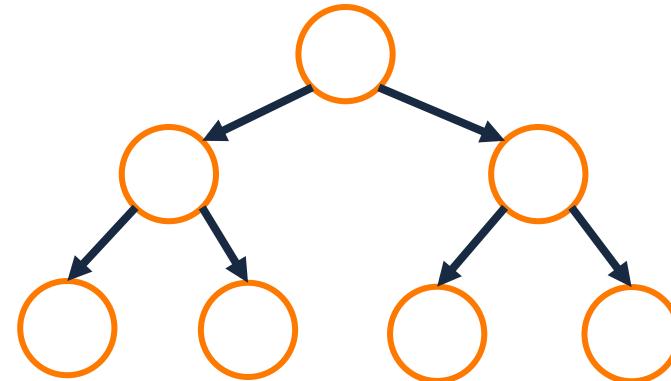
A **binary tree** is a tree where every node has at most two children.



✗ Not a binary tree!

Binary Tree Children

In binary trees, we will label every child as either the “left child” or “right child” of its parent:

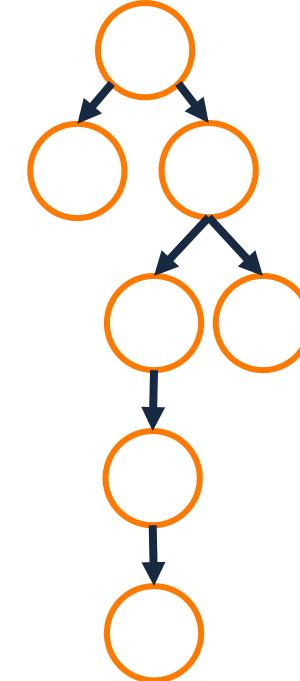
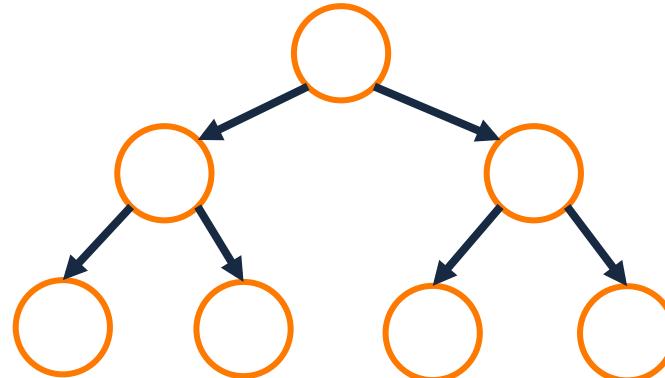


binary-tree/ BinaryTree.h

```
10 template <typename T>
11 class BinaryTree {
12     public:
13         // ...
14
15     private:
16         class TreeNode {
17             public:
18                 T & data;
19                 TreeNode *left, *right;
20                 TreeNode(T & data) :
21                     data(data), left(nullptr), right(nullptr) { }
22         };
23         TreeNode *root_;
24     };
```

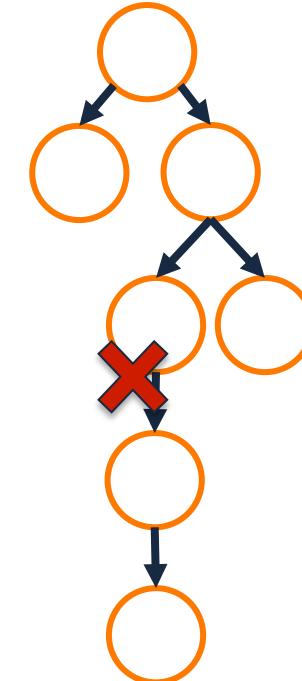
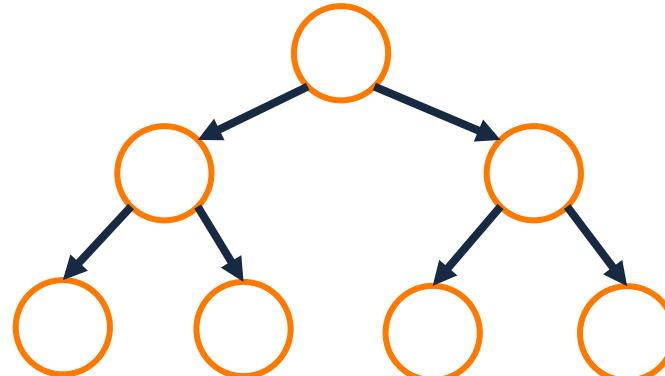
Binary Tree Property: Height

The **height** of a binary tree is the number of edges in the longest path from the root to a leaf.



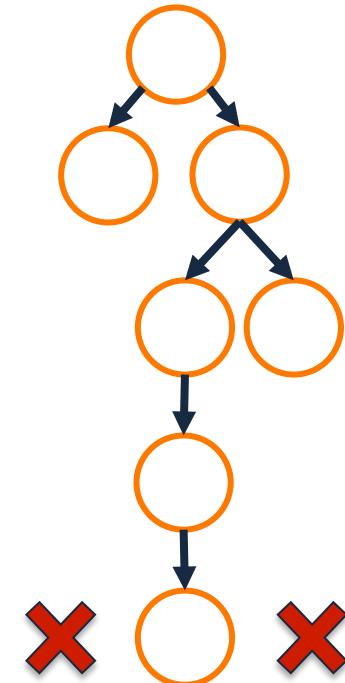
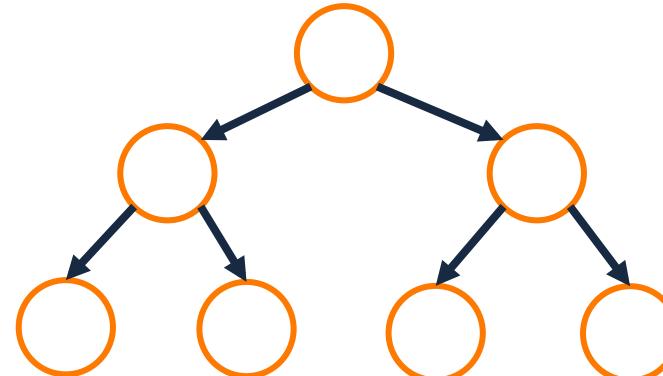
Binary Tree Property: Full

A binary tree is **full** if and only if every node has either zero children or two children.



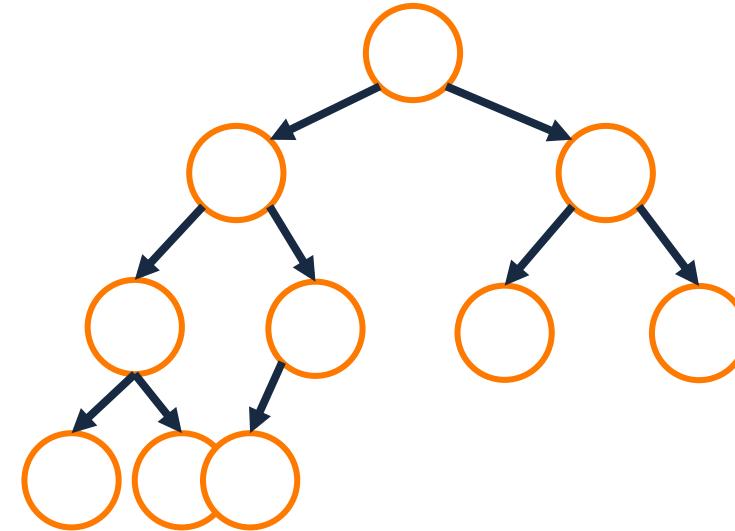
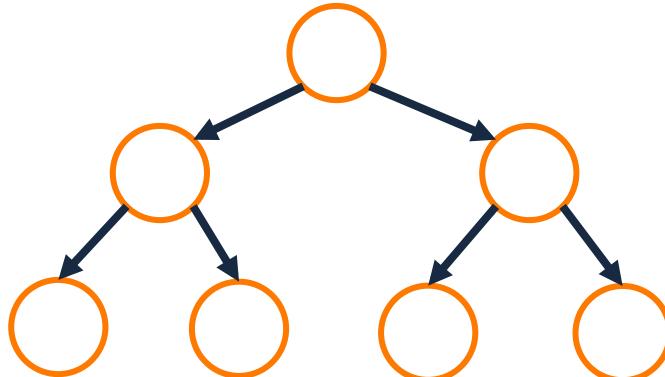
Binary Tree Property: Perfect

A binary tree is perfect if and only if all interior nodes have two children and leaves are at the same level.



Binary Tree Property: Complete

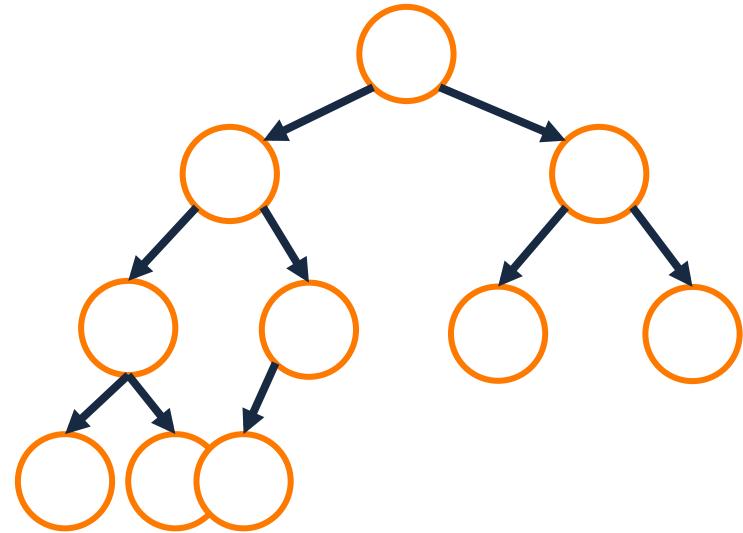
A binary tree is **complete** if and only if the tree is perfect up until the last level and all leaf nodes on the last level are pushed to the left.



Puzzle:

Is a full tree complete?

Is a complete tree full?



Binary Trees

- Binary Trees are a special case of a tree where each node has at most two children.
- The children of binary trees are referred to as the “left child” and “right child”.
- Binary Trees have a **height** and a definition for being **full**, **perfect**, and **complete**.

