Trees

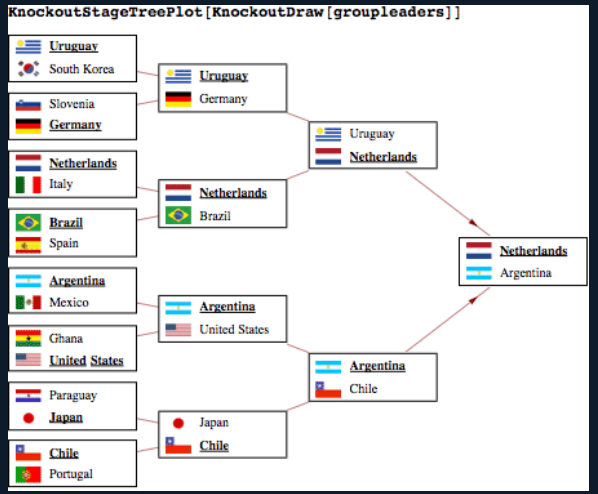
A Graph that is connected and acyclic

Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Contents

- Tree Terminology
- Components of a Tree
- Balanced Trees
- Binary Trees
- Inserting + deleting nodes
- Summary
- References



Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Tree

- A tree is a type of directed graph
  - A minimally connected graph
  - The number of edges in a tree is the number of nodes minus one.
  - Only one path between any two nodes
  - A tree starts with a single “root” node
  - Each node exists only once within the tree

Tree terminology

- Each node (apart from the “root” node) must have one and only one edge leading into it. The node at the other end of that edge is known as the parent

- Each node (including the root) can have multiple edges leading from them. The nodes at the other end of these nodes are known as children.

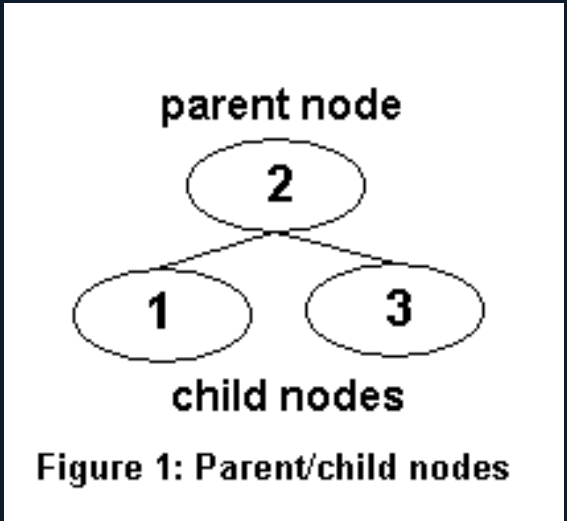


Figure 1: Parent/child nodes

Canberra Institute of Technology

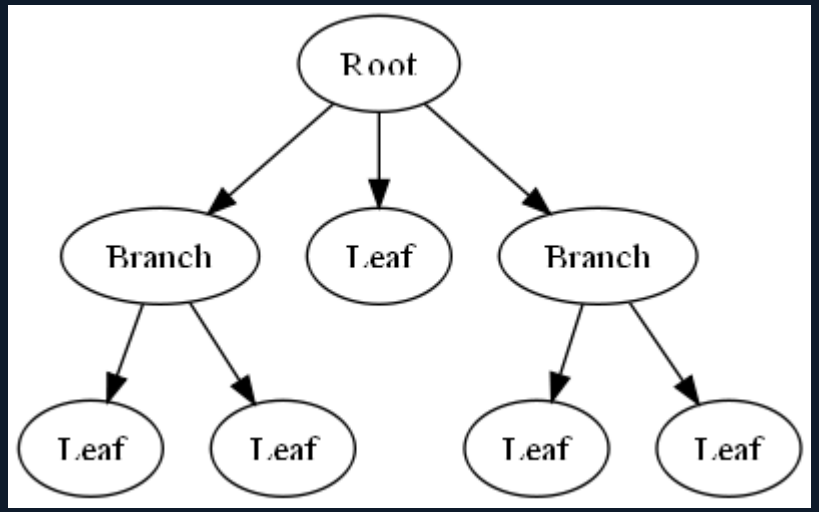
aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

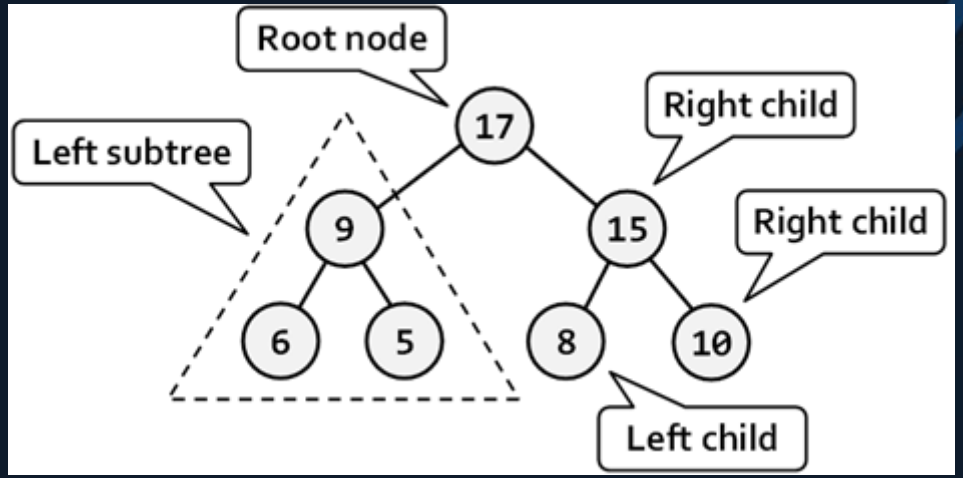
Components of a Tree

- A Tree graph is usually drawn upside down (when compared to a real-world tree)
  - The first node in the tree is the “root” and is located at the very top
  - Nodes without any child nodes are usually referred to as “leaves”
  - Nodes with child nodes are called “branches”
  - A “subtree” is a part of the tree that, if removed, would form a tree on its own.

Components of a Tree

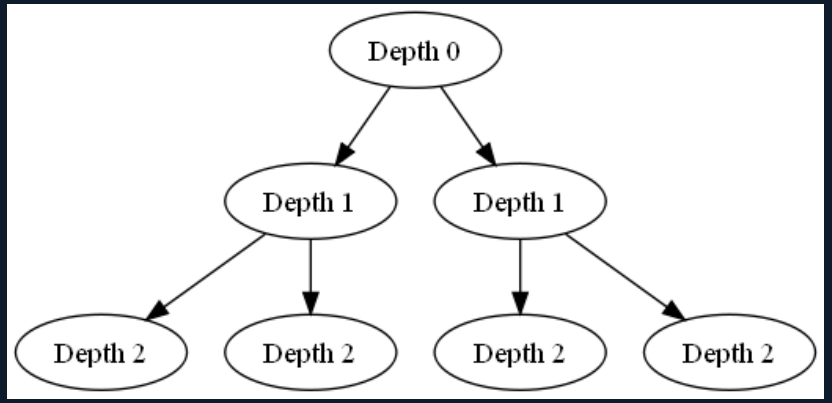
- Image of a Tree





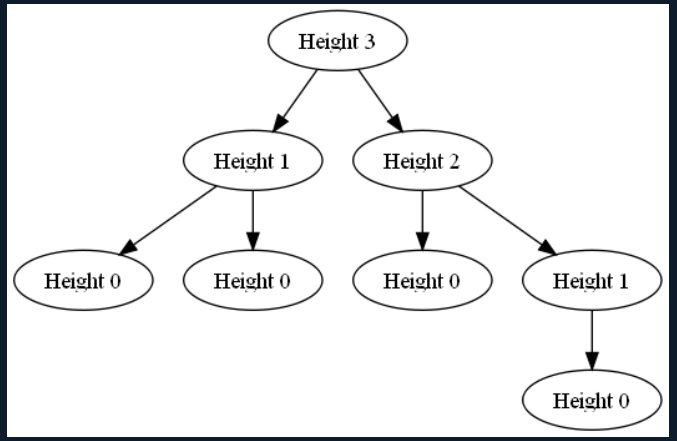
Components of a Tree

- Nodes in a Tree have
  - Depth: the distance of a node from the root node. This is also sometimes referred to as the *level* of a node.

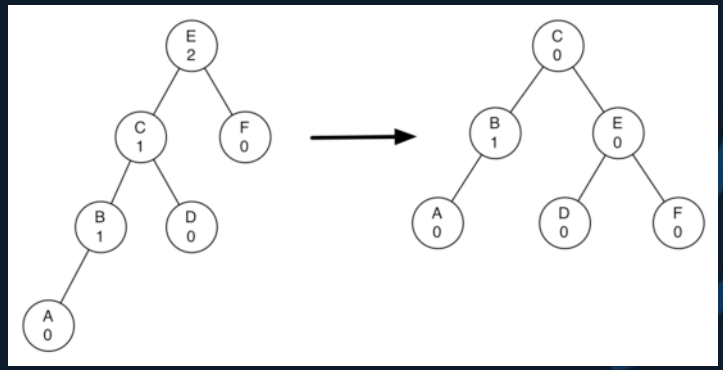


Components of a Tree

- Nodes in a Tree have
  - Height: is the number of nodes from the current node to the deepest leaf node in that branch



Balanced Trees



- A Tree is considered “balanced” when all leaf nodes differ in height by no more than 1.
- Balanced trees are more efficient to search than unbalanced ones.
- There are methods to rebalance a tree, but these can be quite complex.

Canberra Institute of Technology

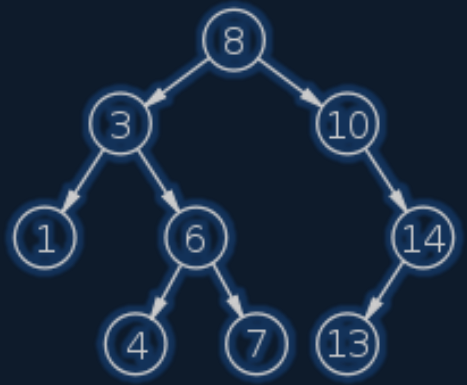
aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Tree Graph Uses

- Like a normal graph, a tree's main use is not for storage
- Trees are a great way for representing hierarchical data
  - The bones in a skeleton in computer graphics
  - Spatial layout of a game level
    - The entire level is under the root node, divided into subtrees representing areas of the level, each with more subtrees representing the game entities within those areas
  - XML documents
  - Scene Graph – Parent/Child hierarchies within objects

Binary Trees

- A tree where each nodes has a maximum of 2 children
  - The left subtree contains key values less than the root
  - The right subtree contains key values greater than the root



Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Binary Trees - Node Structure

- The node itself is very similar to a graph node.
- There is no need to store edges. We just need to store pointers to child nodes.
  - For a binary tree we just store the two pointers. For more complex tree we might need a list or array of pointers.

```
class TreeNode
{
private:
    //this could also be a pointer to another object if you like
    int value;

    //Node's children
    TreeNode* left;
    TreeNode* right;
};
```

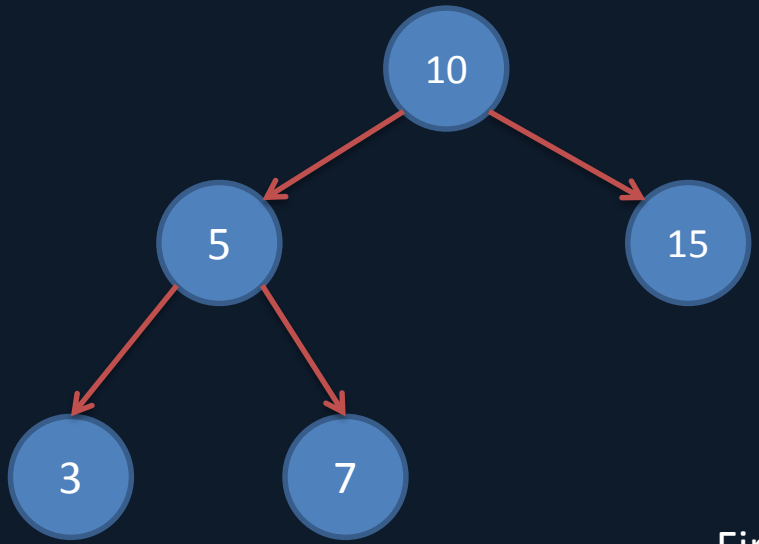
Canberra Institute of Technology

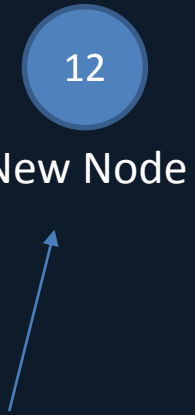
aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Binary Trees - Inserting

- Inserting in a Binary Tree is easy!
  - Create a new node to store our new value.
  - Compare the value of our new node to the root, if the new value is lower, check the left child. If higher, check the right child.
  - Compare the new value to the child, and again if lower check left, if higher check right.
  - Keep going until we find an empty leaf, place the new node there.

Binary Trees - Inserting



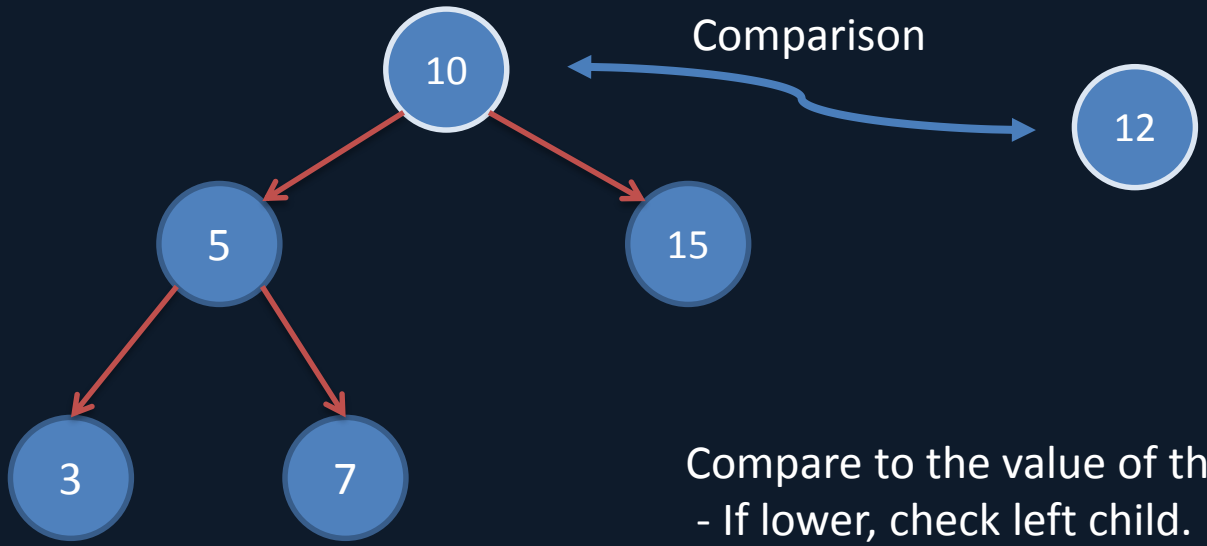


First, we create our new node

Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Binary Trees - Inserting

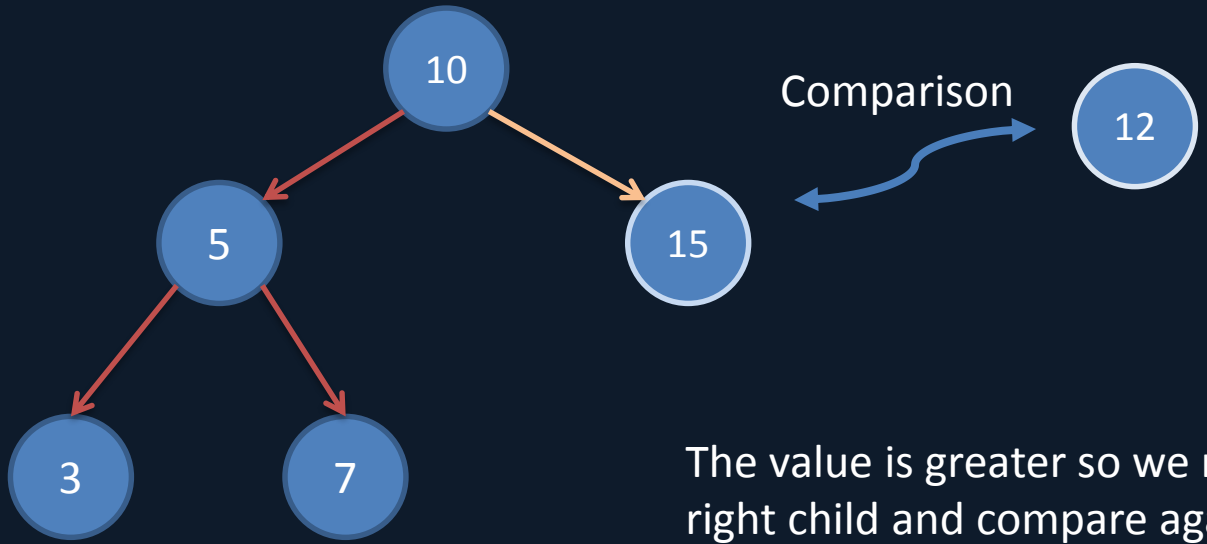


Compare to the value of the root:
- If lower, check left child.
- if greater, check right child.

Canberra Institute of Technology

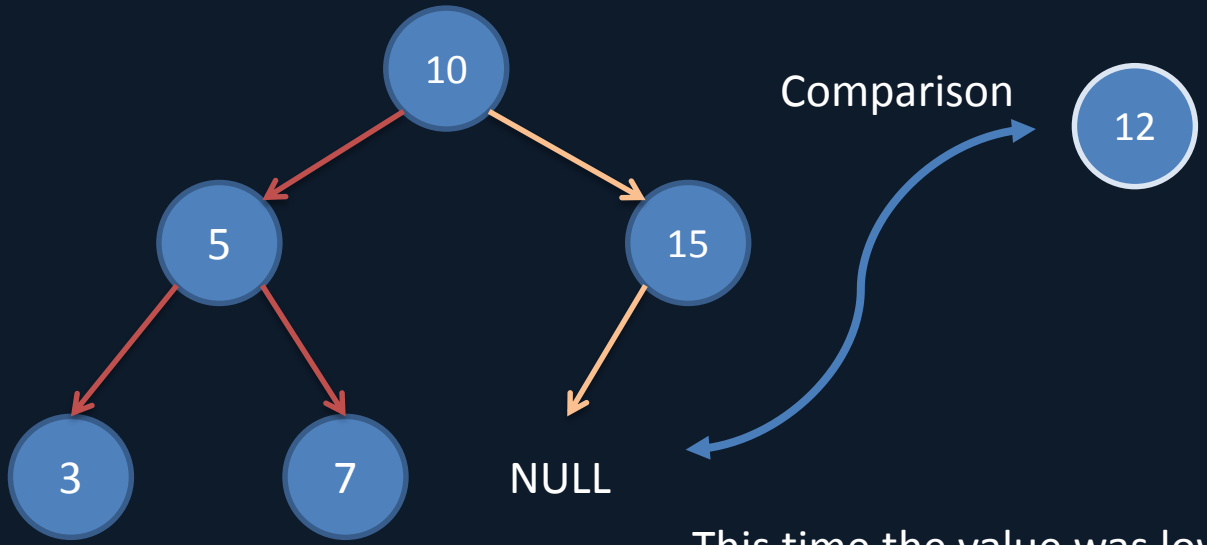
aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Binary Trees - Inserting



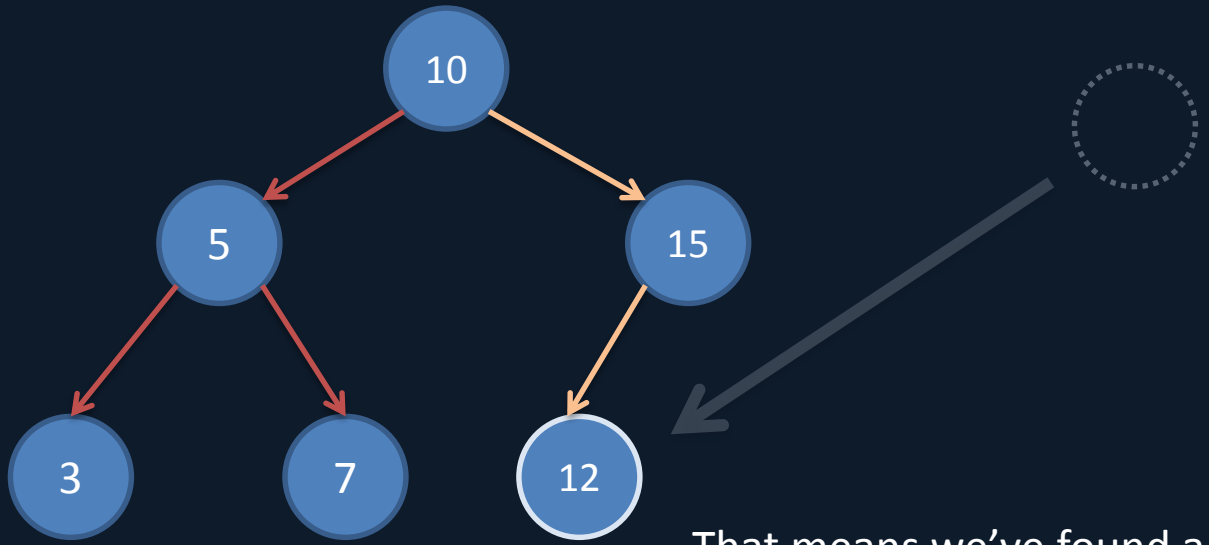
The value is greater so we move to the right child and compare again.

Binary Trees - Inserting



This time the value was lower, so we move left… and run into a null pointer.

Binary Trees - Inserting

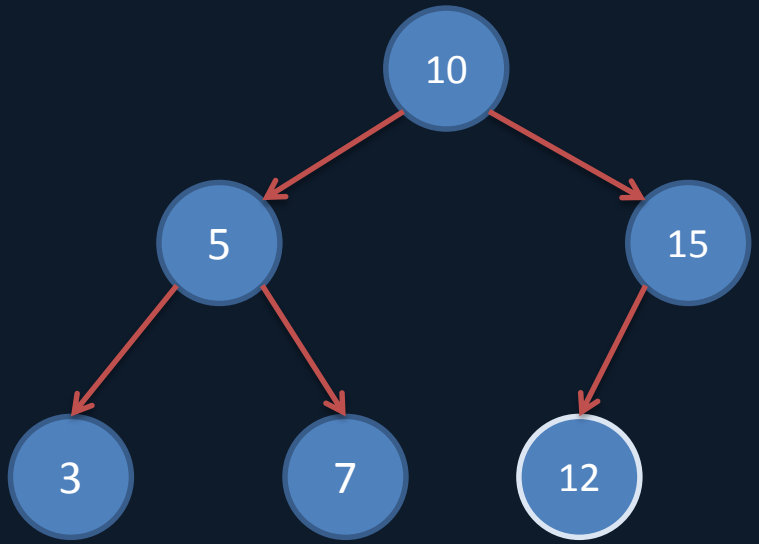


That means we've found a valid spot and we insert our node.

Binary Trees - Deleting Nodes

- Deleting nodes can be quite difficult. There are three different situations we need to deal with.
  - Deleting a leaf.
  - Deleting a node with one child.
  - Deleting a node with two children.

Binary Trees - Deleting Nodes

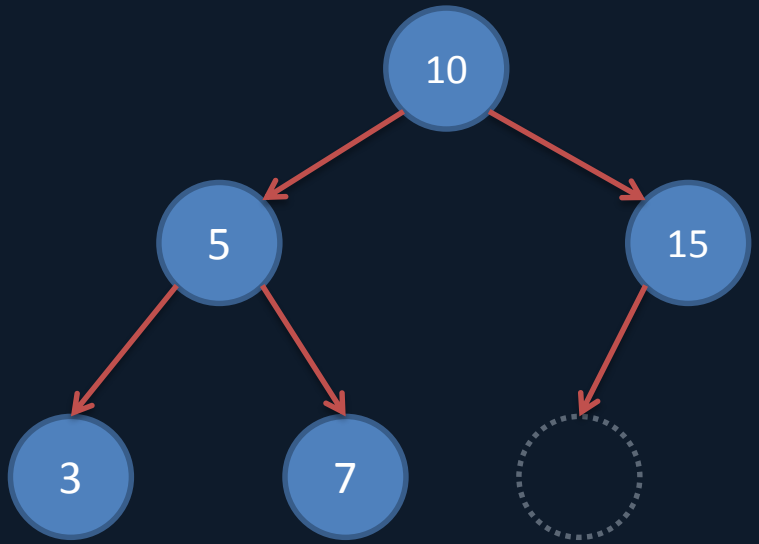


Deleting a leaf is easy.
If we want to remove node 12…

Canberra Institute of Technology

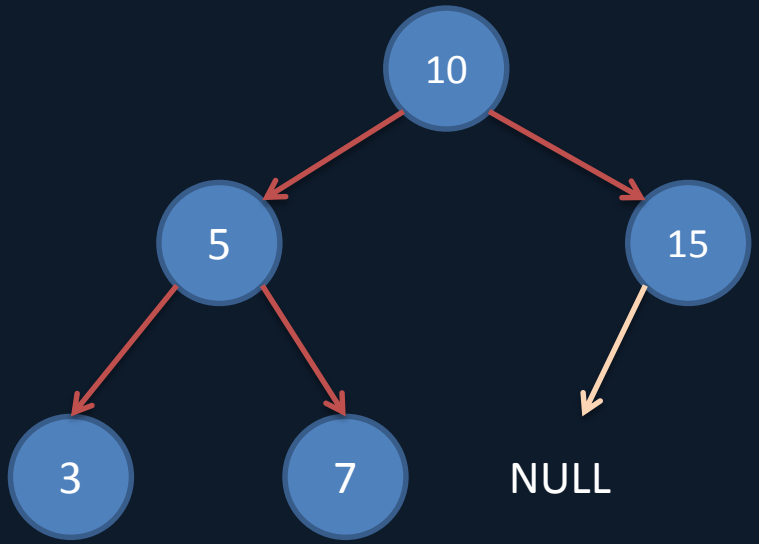
aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Binary Trees - Deleting Nodes



We simply delete it…

Binary Trees - Deleting Nodes



Then set the parent's pointer to it to null and we're done.

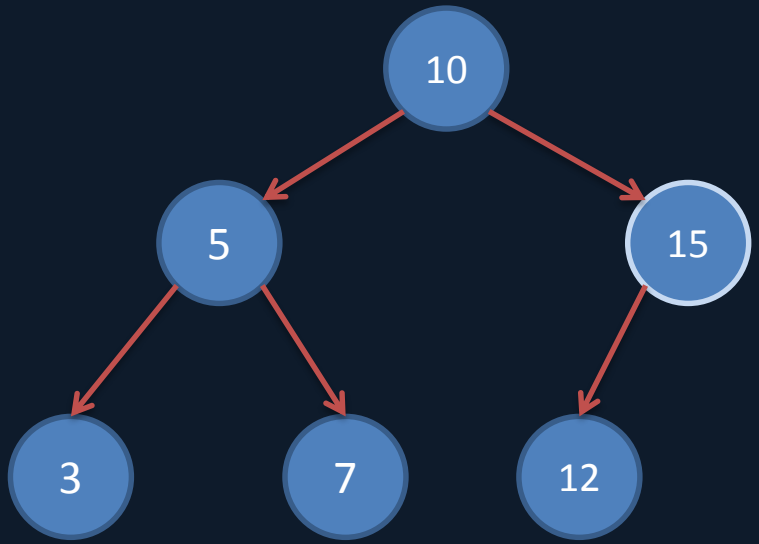
Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Binary Trees - Deleting Nodes

- Deleting a node with one child is still pretty easy.
  - We just need to deal with that node's one child to make sure we don't lose it.

Binary Trees - Deleting Nodes

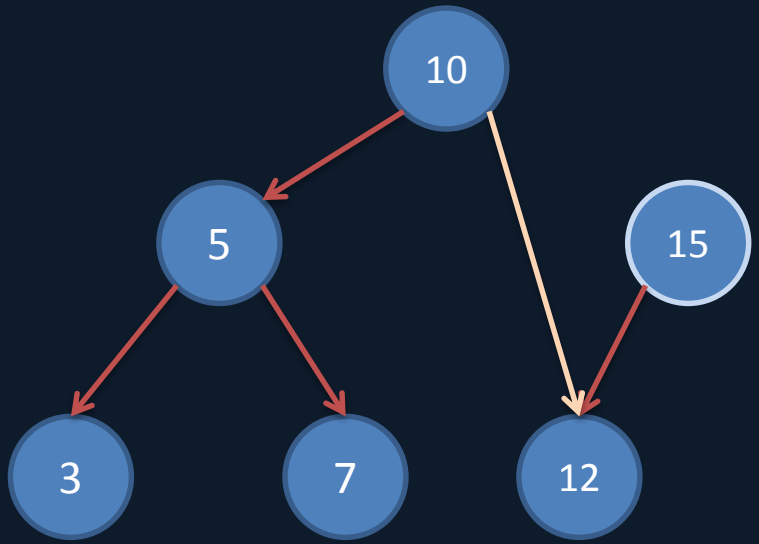


Lets say we want to remove node 15.

Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Binary Trees - Deleting Nodes

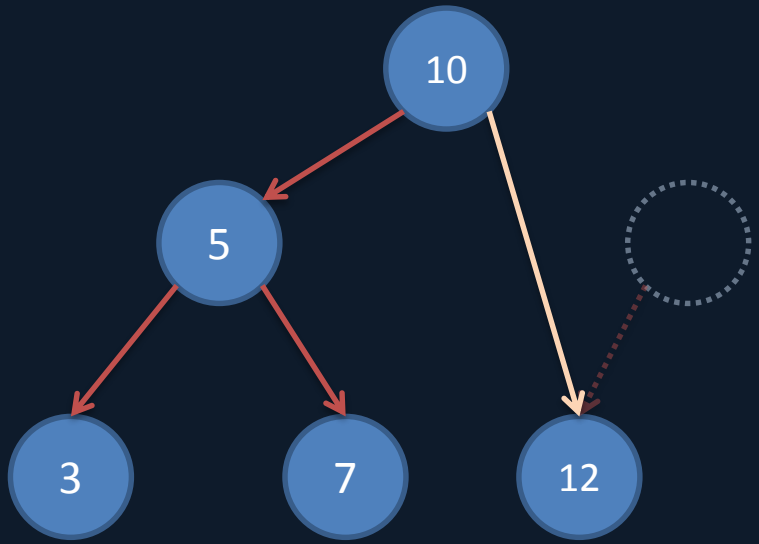


First we change its parent's pointer to it so that it instead points to its child.

Canberra Institute of Technology

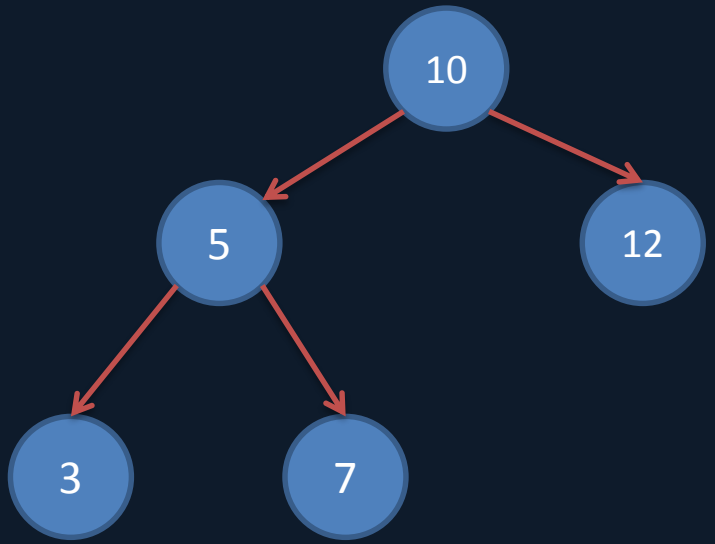
aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Binary Trees - Deleting Nodes



Then we just delete the node…

Binary Trees - Deleting Nodes

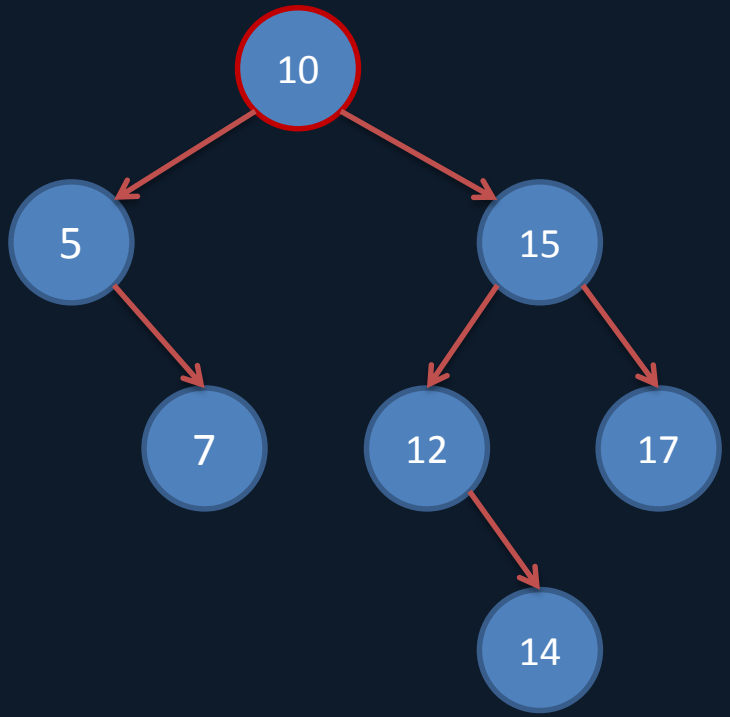


And we're done.

Binary Trees - Deleting Nodes

- Deleting a node with two children is hard!
  - We not only need to keep both children, we need to reorder the tree to keep the values correct.
  - This is difficult and inefficient so instead we cheat and do a sneaky swap…

Binary Trees - Deleting Nodes



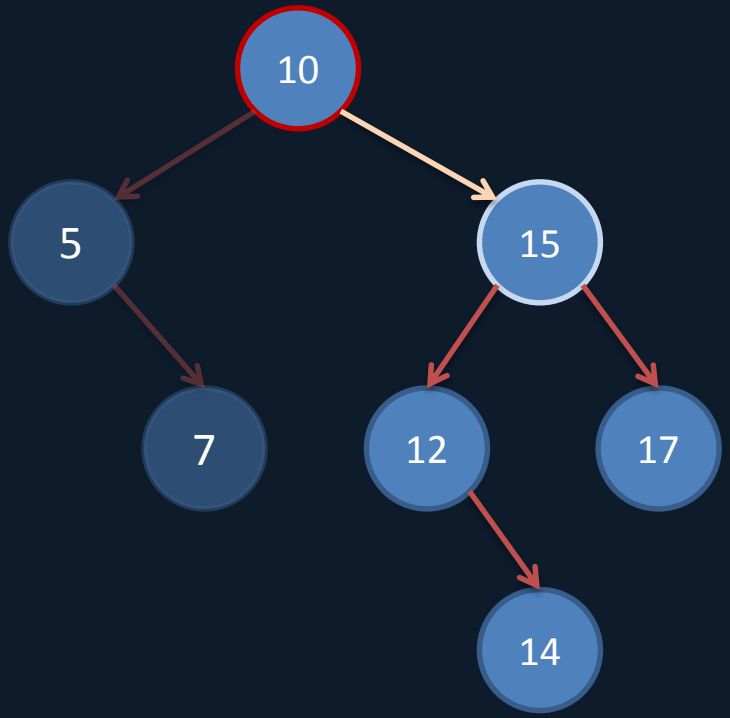
Lets say we want to remove node 10 from this tree.

This time we're marking the target node in red to avoid later confusion.

Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Binary Trees - Deleting Nodes



We need to find the smallest value in the tree that is greater than the one we want to remove.

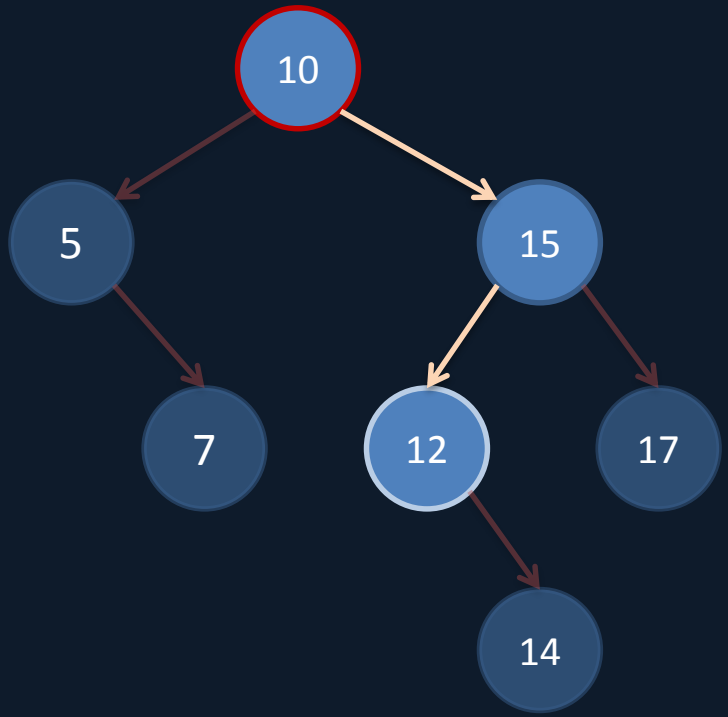
To do this, first we step right once so that now we are only dealing with values greater than the node we are removing.

We know all values to the left are lower so we don't need to touch any of those.

Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Binary Trees - Deleting Nodes



Now that we are only dealing with values greater than our target node, we need to find which is the smallest.

To do this we step left as many times as we can until we run out of nodes.

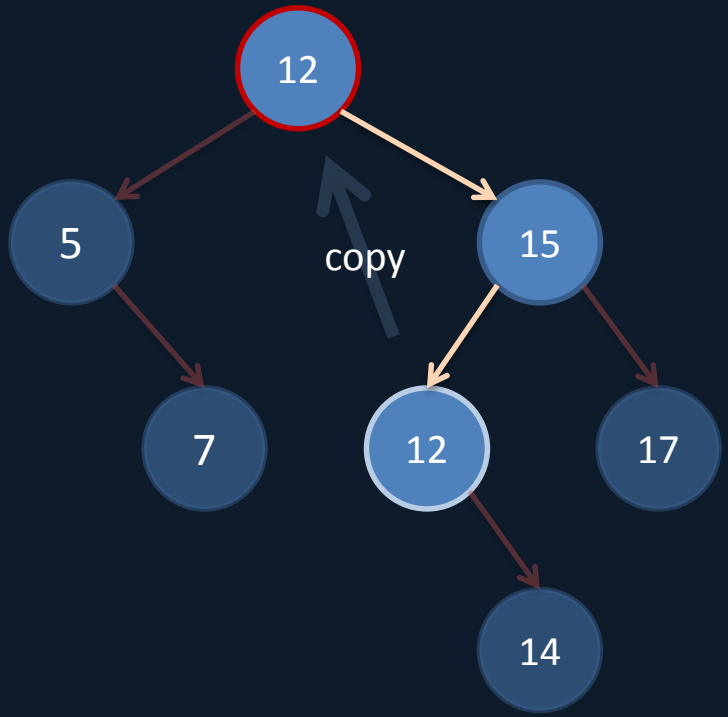
In this case we find it's node 12.

We won't be touching any nodes that would be reached by stepping right again since those would be greater.

Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Binary Trees - Deleting Nodes



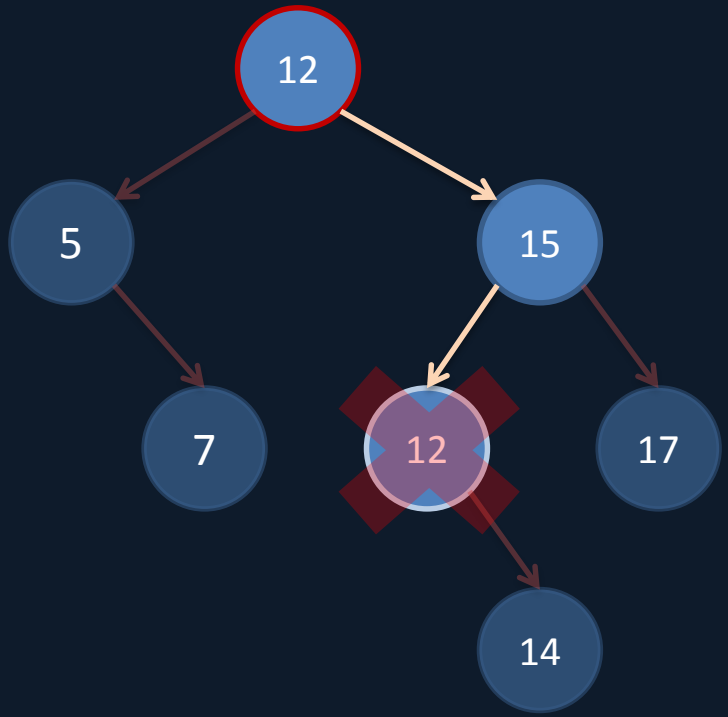
The next step is to copy the value of the node we found (12) over the red node we want to remove.

So now the red node's original value has been discarded and we have two nodes with the same value.

Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Binary Trees - Deleting Nodes



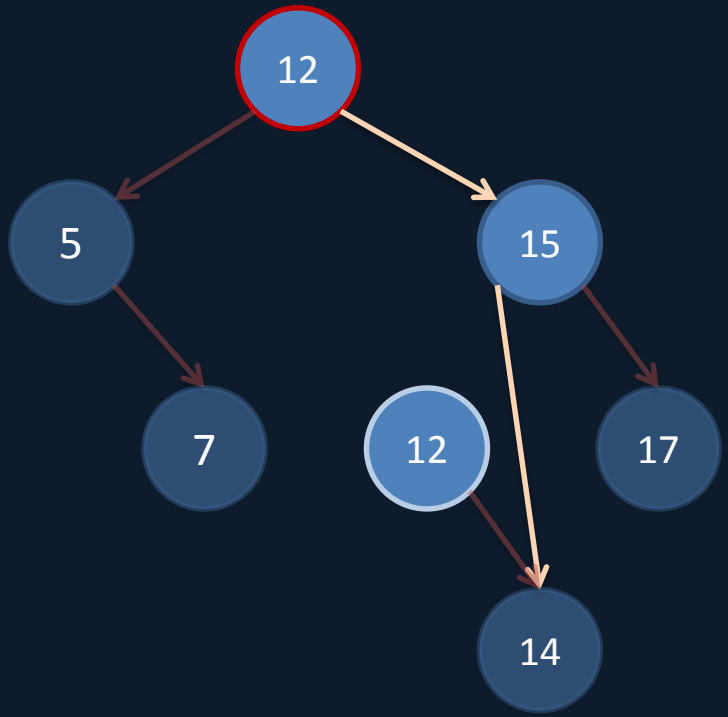
Now we want to delete the original node 12 since its value has been relocated to the red node.

We do this using one of the two methods discussed previously: Either it has no children and we can simply remove it, or it has one and we need to fix up some pointers.

Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Binary Trees - Deleting Nodes

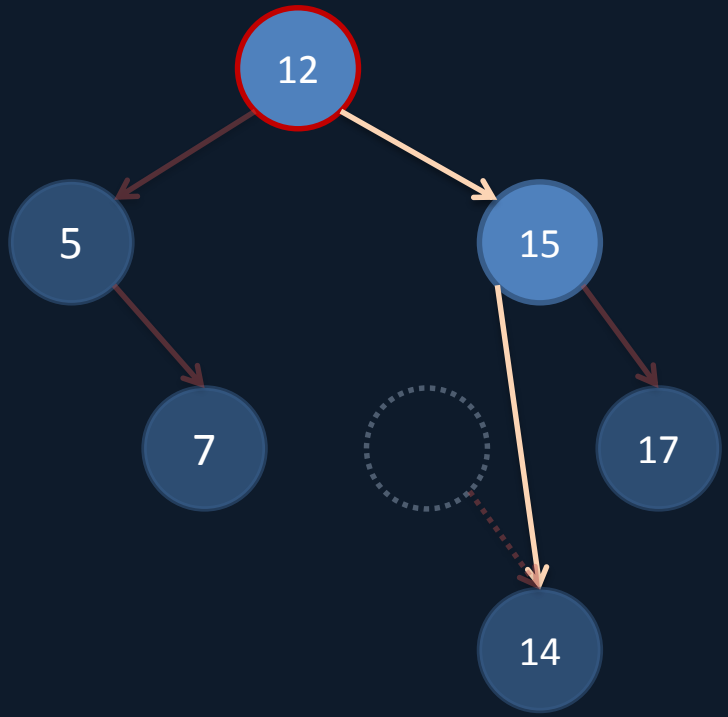


In this case it has one child so we change its parent's pointer so that it instead points to its child.

Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Binary Trees - Deleting Nodes

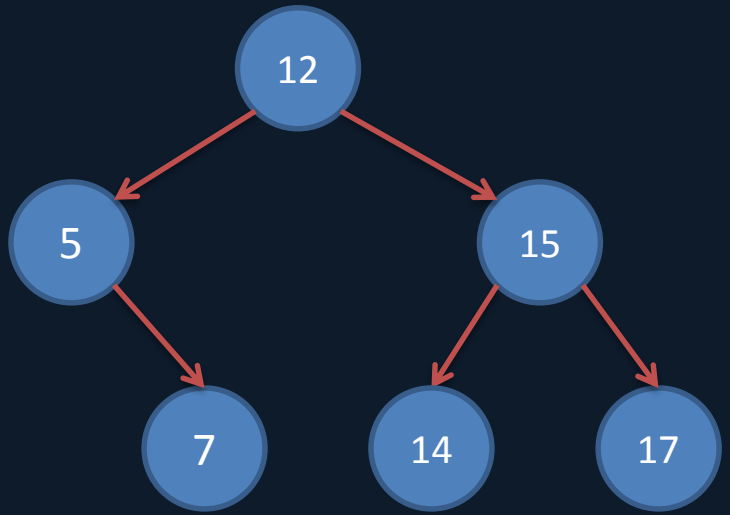


Then we delete it.

Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Binary Trees - Deleting Nodes



… and finally we're done.

Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Summary

- Binary Trees are just one simple type of tree, there are many others.
- Trees are an extremely useful tool in programming
  - They serve as flexible data structures
  - Allow data to be accessed and manipulated
- Now we are going to write our own implementation in C++

References

- “Data Structures – A Pseudocode Approach with C++”, Richard F. Gilberg and Behrouz A. Forouzan, 2001, Brooks/Cole, Chapter 7: Introduction to Trees

- “Data Structures and Algorithms for Game Developers”, Allen Sherrod, 2007, Delmar Cengage Learning, Chapter 9: Trees