More Binary Search Trees, Project 2

Bryce Boe

2013/08/06

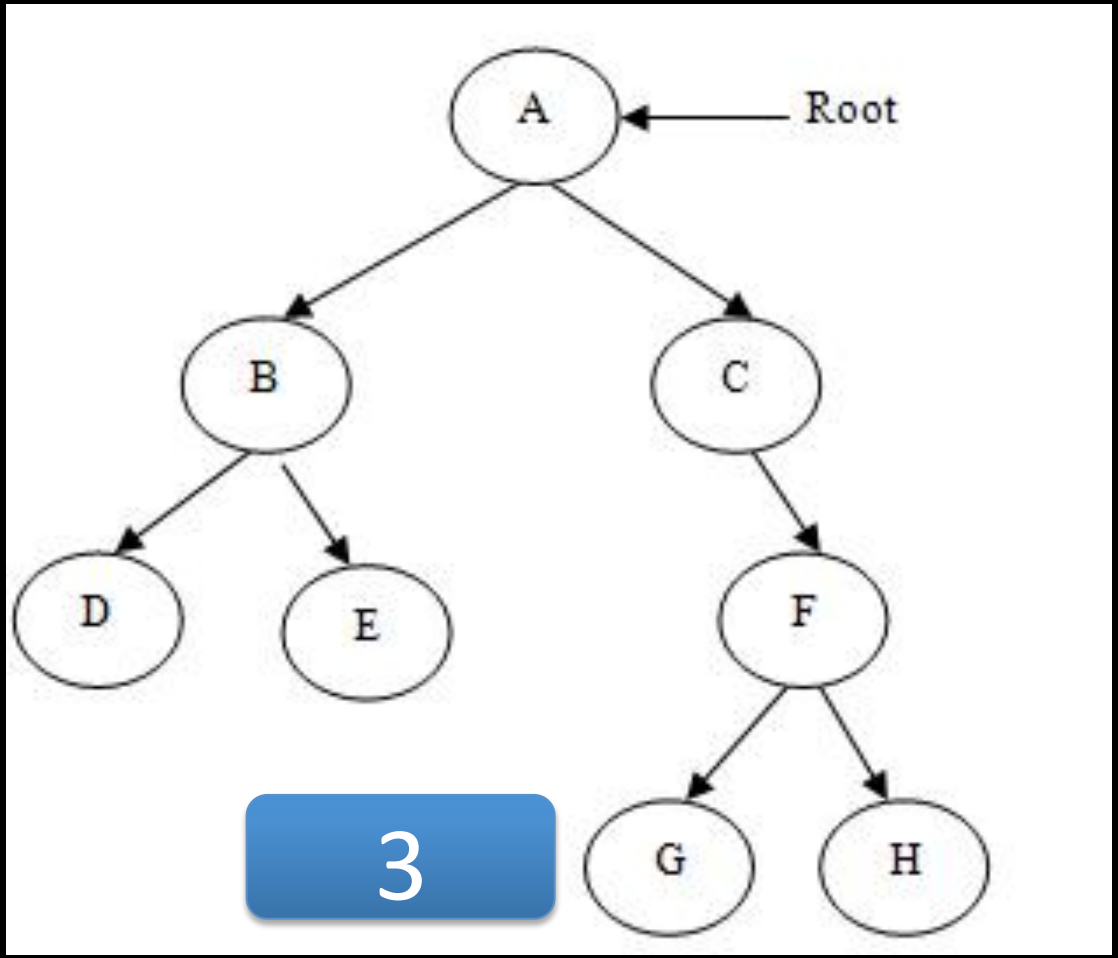
CS24, Summer 2013 C

Outline

- Lab 5 Solution
- Tree Traversals
- More Binary Search Trees
- Project 2

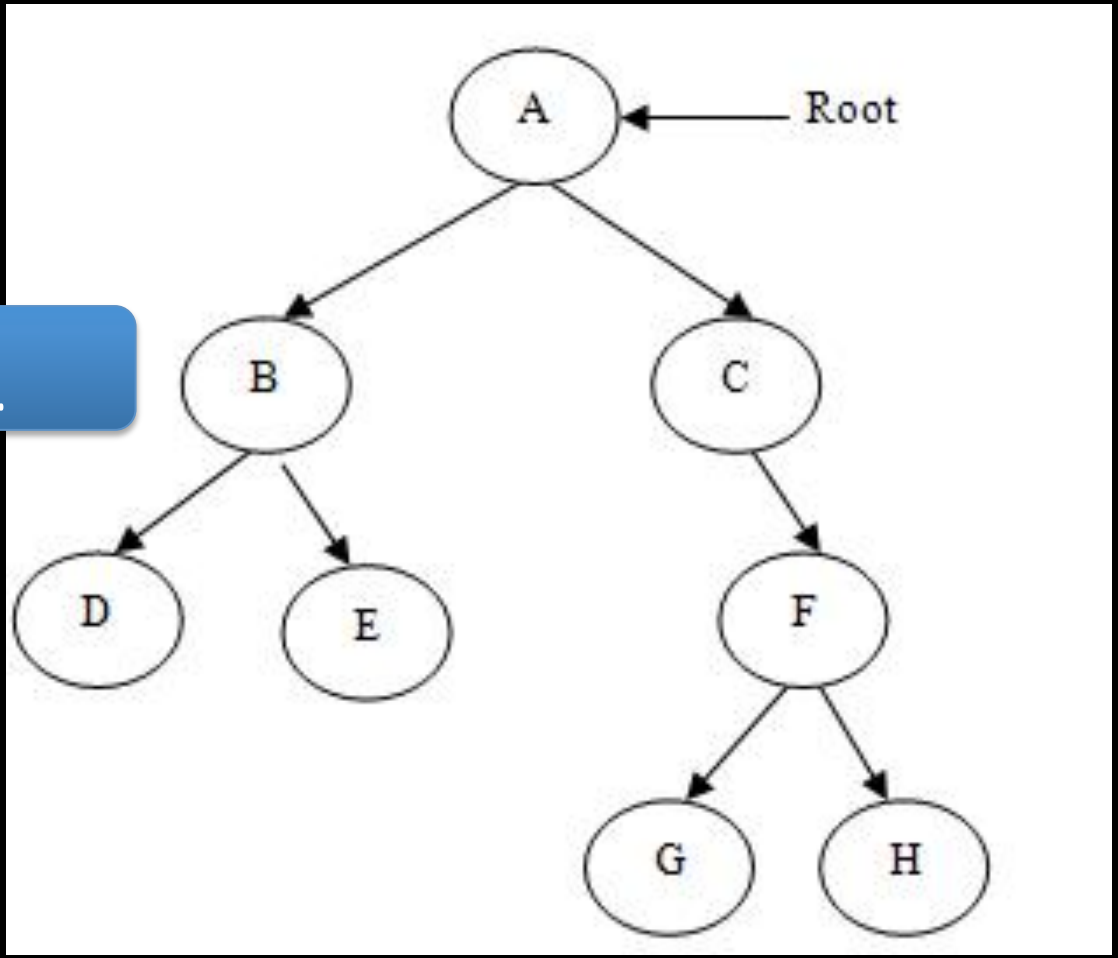
THURSDAY RECAP

What is the depth of G?

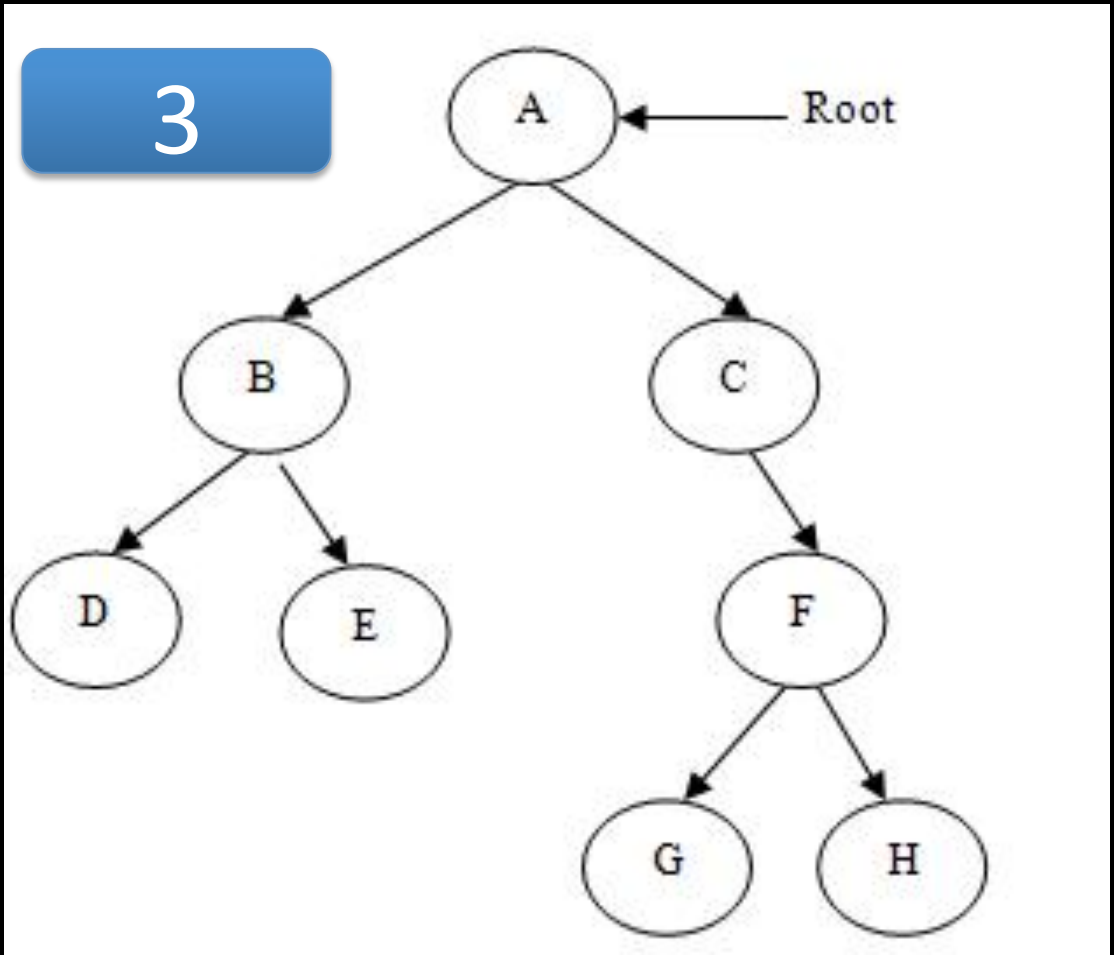


What is the height of B?

1



What is the height of the tree?



Recursion question

- How many activation records are created when calling Fibonacci(0)? 1
- Fibonacci(1)? 1
- Fibonacci(2)? 3
- Fibonacci(3)? 5
- Fibonacci(4)? 9
- Fibonacci(5)? 15

Lab 5 Solution

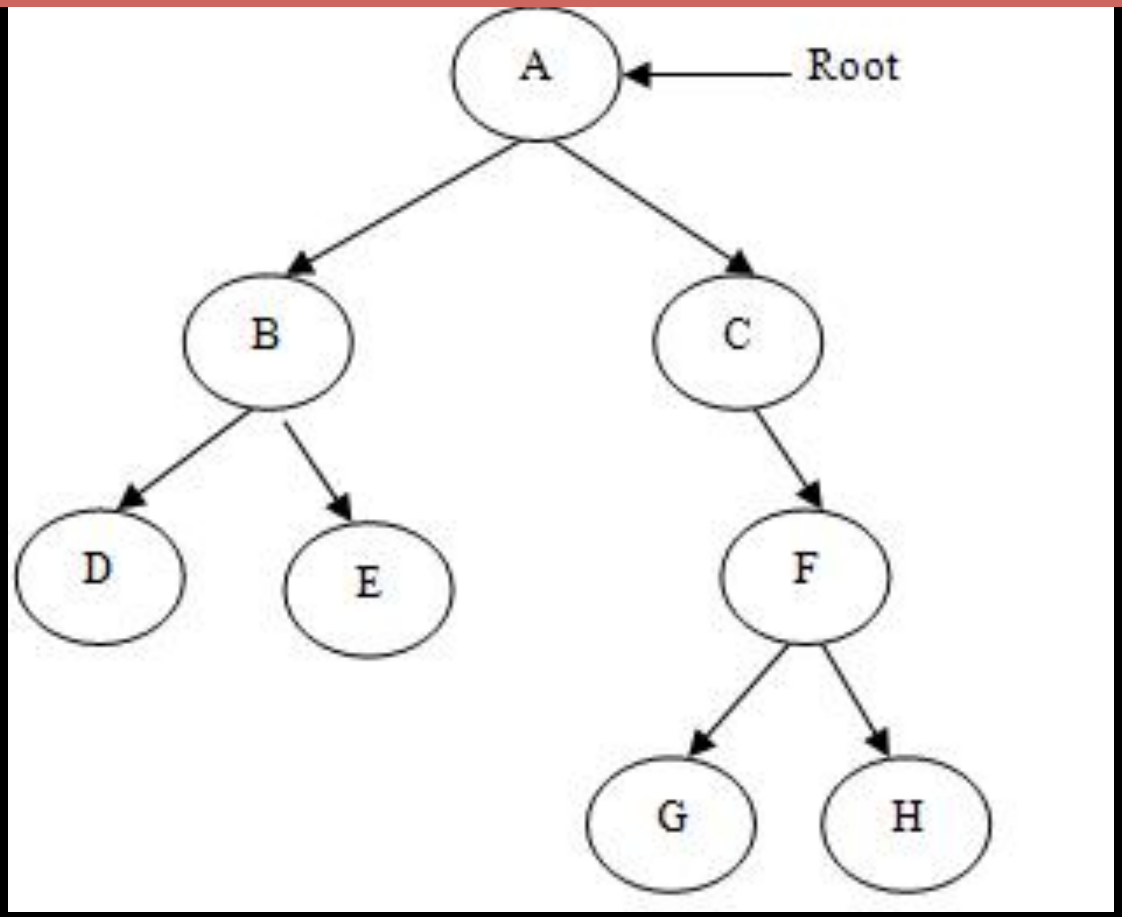
- Going over in class
- Make private request for bst.cpp if you need a 100% working solution

Depth-First Tree Traversals

- Can be done iteratively (with a stack) or recursively
- Pre-order
  - *Process* the node, recurse on left subtree, recurse on right subtree
- In-order
  - Recurse on the left subtree, *process* the node, recurse on the right subtree
- Post-order
  - Recurse on the left subtree, recurse on the right subtree, *process* the node

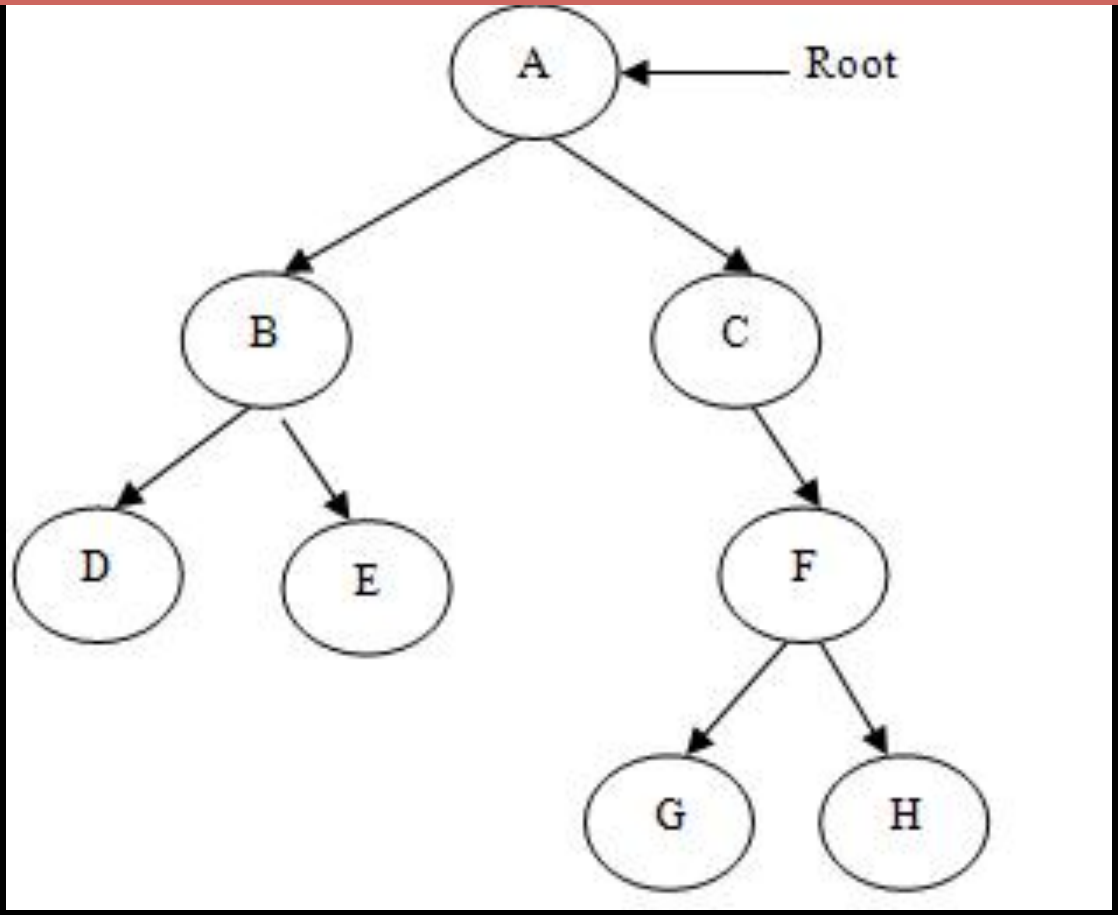
Pre-Order Traversal

A -> B - > D -> E -> C -> F -> G -> H



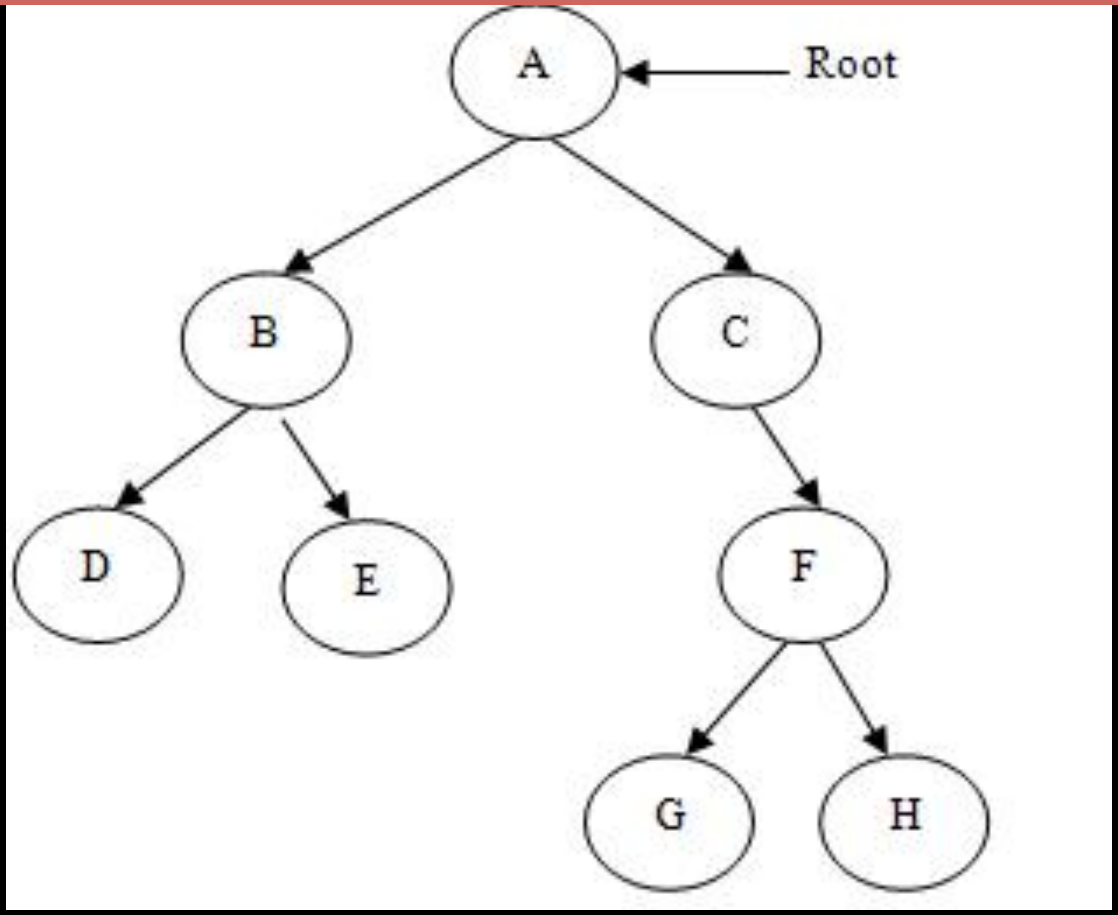
In-Order Traversal

D -> B -> E -> A -> C -> G -> F -> H



Post-Order Traversal

D -> E -> B -> G -> H -> F -> C -> A

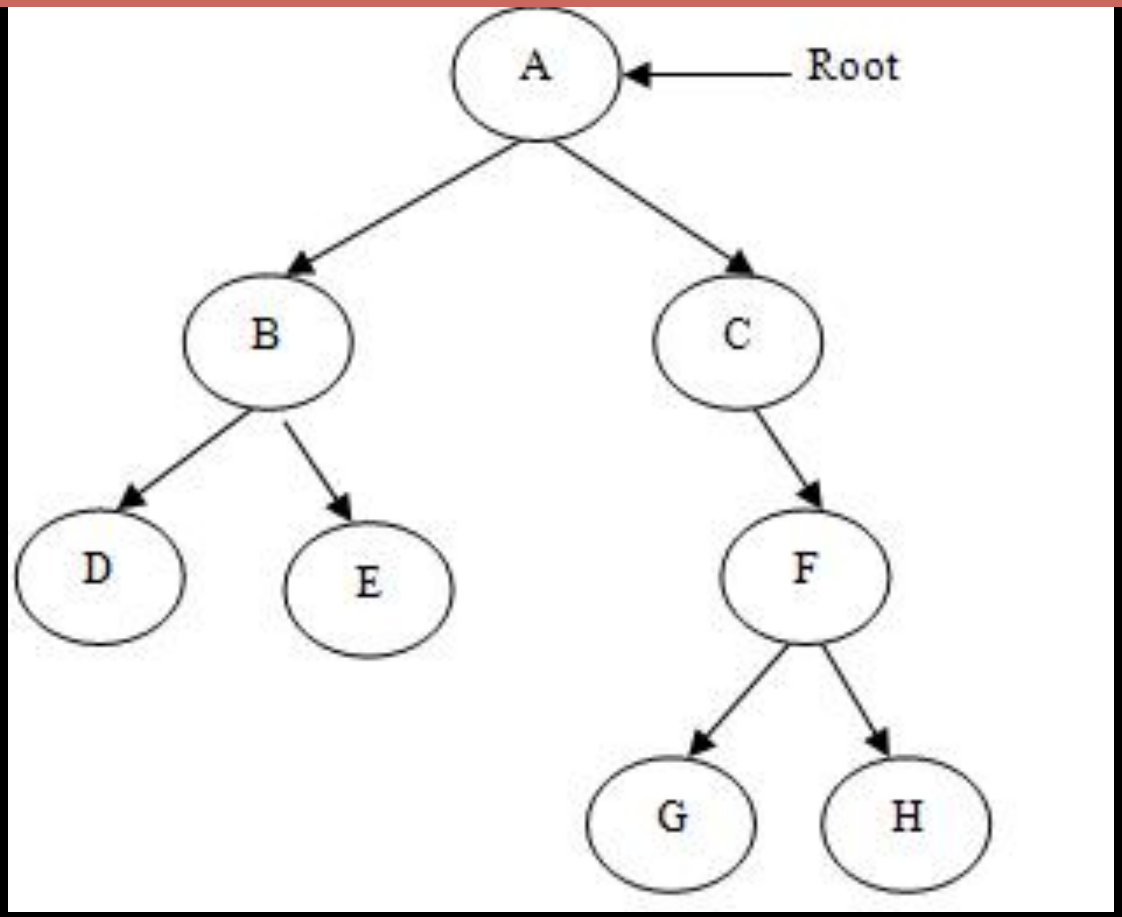


Breadth-first Traversal

- Cannot be done recursively
- Done iteratively with the help of a queue

Breadth-first (queue lhs before rhs)

A -> B -> C -> D -> E -> F -> G -> H



Question

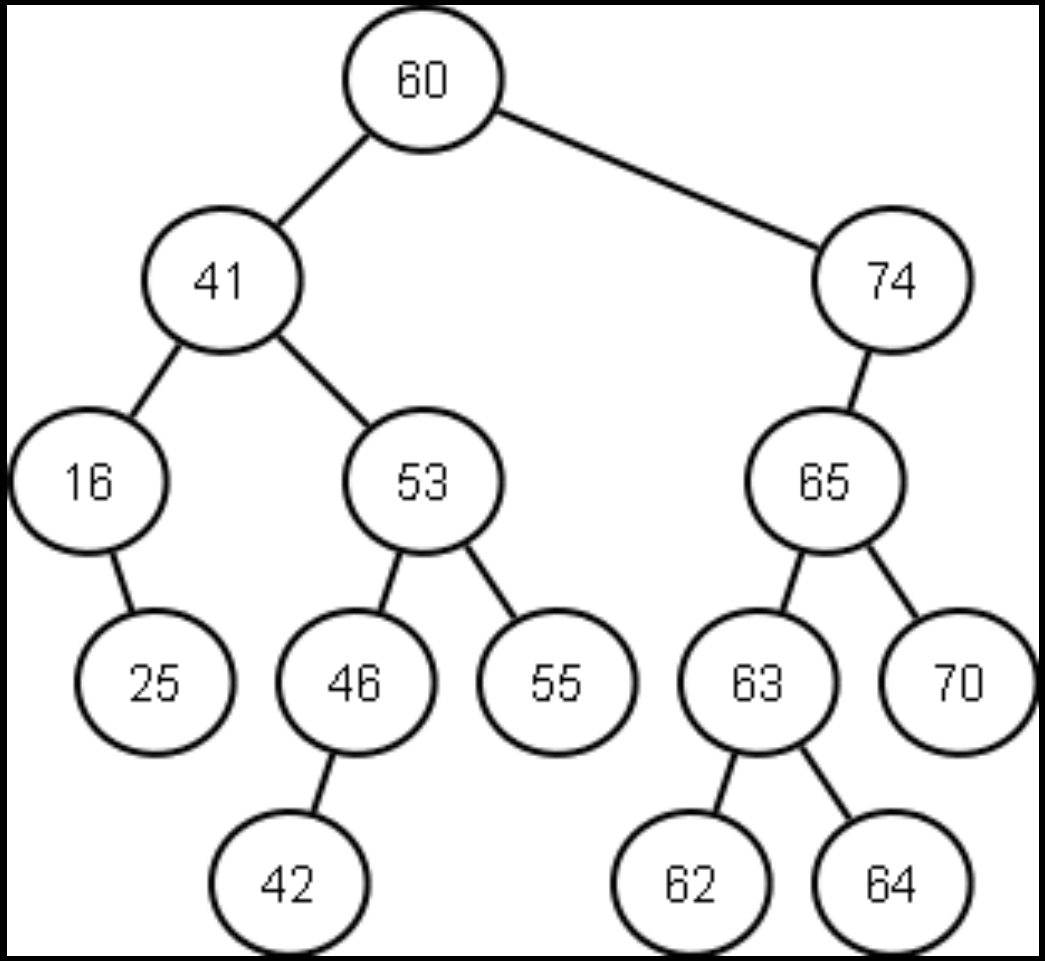
- Which of the traversals will output a sorted version of a binary search tree?

MORE BINARY SEARCH TREES

Binary Search Trees

- Recall
- A binary search tree is a tree with the property that the value of all descendants of a node's left subtree are smaller, and the value of all descendants of a node's right subtree are larger

BST Example



BST Operations

- insert(item) – done in Lab 5
  - Add an item to the BST
- remove(item) – to complete in project 2
  - Remove an item from the BST
- contains(item) – done in Lab 5
  - Test whether or not the item is in the tree

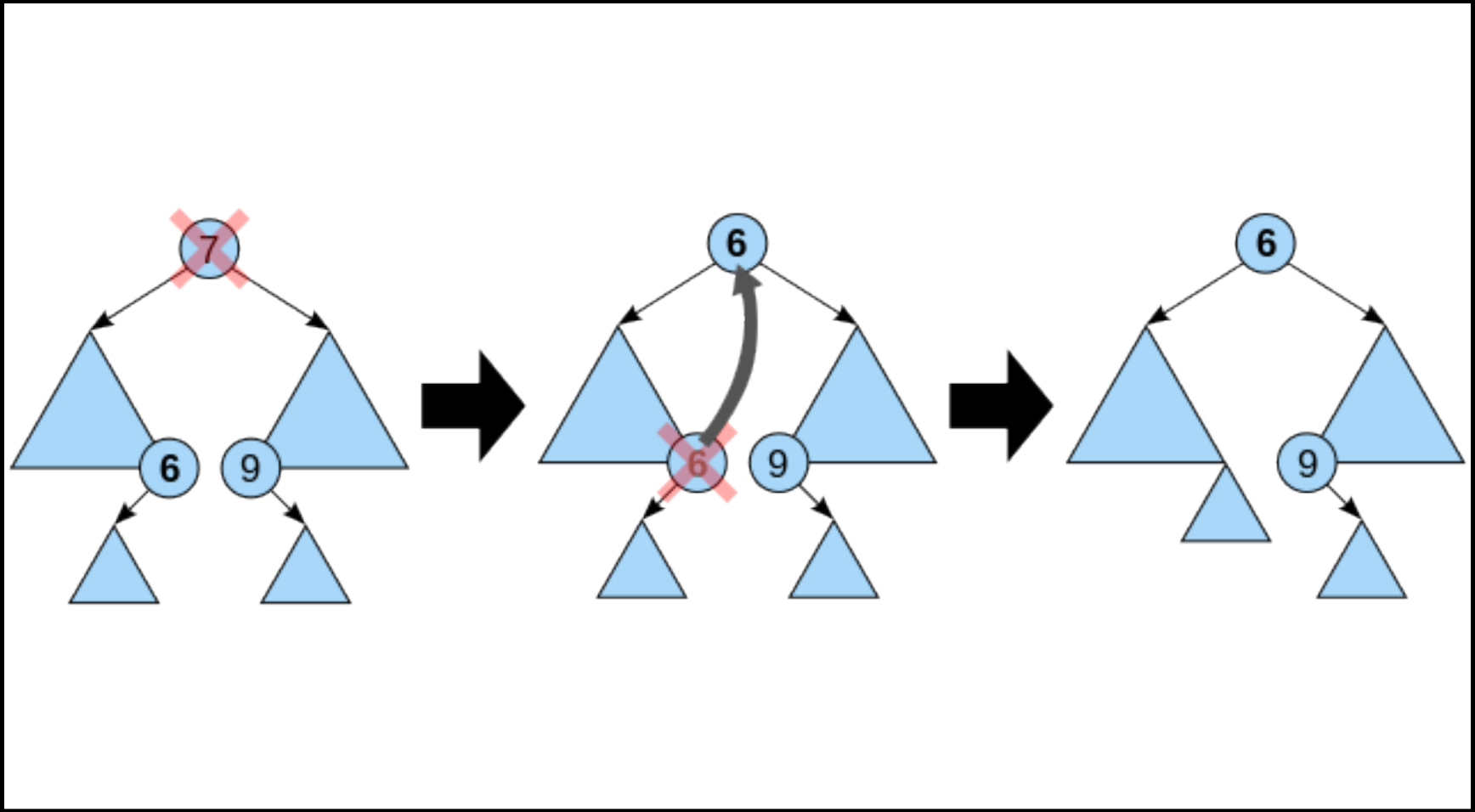
BST Remove

- If the node has no children simply remove it
- If the node has a single child, update its parent pointer to point to its child and remove the node

Removing a node with two children

- Replace the value of the node with the largest value in its left-subtree (right-most descendant on the left hand side)
- Then repeat the remove procedure to remove the node whose value was used in the replacement

Removing a node with two children



PROJECT 2: VIRTUAL DIRECTORY TREE