CS 10:
Problem solving via Object Oriented Programming

Winter 2017

Tim Pierson
260 (255) Sudikoff

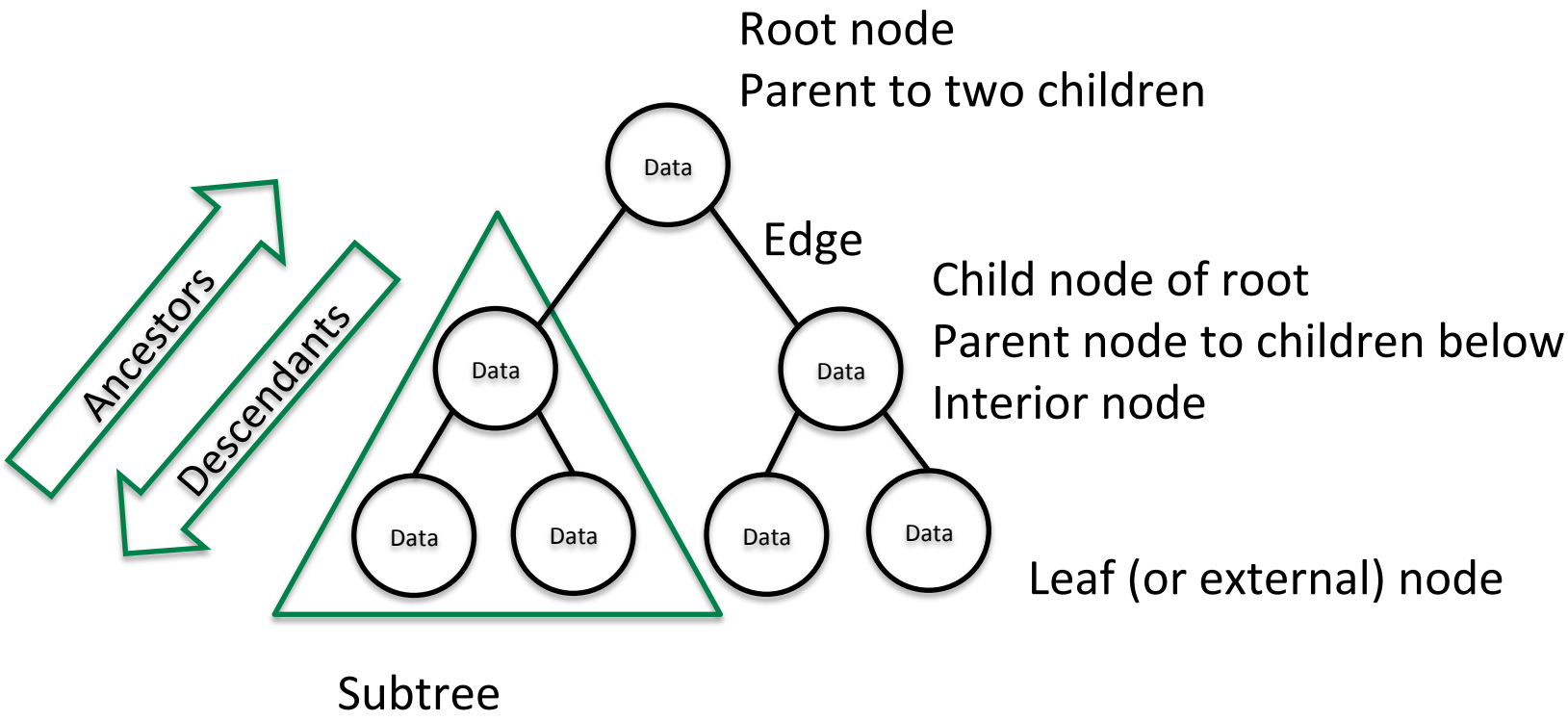
Day 8 – Hierarchies

Agenda

1. General-purpose binary trees
2. Tree traversal
3. Expression trees

We can represent hierarchical data using a data structure called a tree

Tree data structure



Binary trees nodes have 0, 1, or 2 children

BinaryTree.java

- Each node has a left and right child (those could have children or could be null) and data (generic type E for element)
- Simple accessors to determine isInner, isLeaf, left child, right child
- No parent edge, but would be easy to add one
- Generally will start at root and work downward
- Different from linked list because list has 0 or 1 other nodes, tree has 0, 1, or 2, so can't just proceed forward in a loop
- Consequently most tree code is recursive

To calculate size() from any given node, must calculate size of children +1

BinaryTree.java

- *size()*
  - returns number of nodes in a tree, or subtree if not starting at root
  - Size is size of left subtree + size of right substree + 1 (this node)
  - Makes recursive calls to *size()* method of children
  - Those children will in turn call their children for size
  - Until we hit leaf which has not children, so no more calls to children's *size()* (could have explicitly check for *isLeaf()* but don't have to)
  - At that point *num* = 1, and we recurse back up tree

Height and equals

BinaryTree.java

- *height()*
  - returns longer of two paths down left and right, plus 1
  - Leaf nodes are of height 0

- *equals()*
  - Returns true if two trees are equal
  - Must check same number of nodes and each node contains the same data
  - Why can't we just use == on each node? (because only == if point to same memory location, Java doesn't know how to compare tree nodes)

fringe() uses an accumulator to avoid inefficiencies

BinaryTree.java

- *fringe()*
  - Returns leaves in order, left to right
  - Could append fringe of left to fringe of right, but ArrayList append ends up copying one of them, would end up with quadratic time algorithm as stuff gets copied again and again
  - Accumulator pattern saves the day!
  - Pre-create an empty list *f* (the accumulator) and pass it to helper function *addToFringe(f)*

- *addToFringe(ArrayList<E> fringe)*
  - If leaf node, add *data* (type is E) to *f*
  - Else, if *hasLeft()*, *left.addToFringe(f),* if *hasRight(), right.addToFringe(f)*
  - *f* "accumlates" the fringe

toString() also uses an accumulator to create a string representation of the tree

BinaryTree.java

- *toString()*
  - Many ways to print a tree
  - Here we print it in hierarchical structure like a chapter/ section/subsection in a book, indenting by 2 spaces
  - Start with empty string accumulator, then call helper

- *toStringHelper(String indent)*
  - Create a new String *res = indent + data + new line (holds this node's data, now get children)*
  - *Build children*
  - If *hasLeft() res += left.toStringHelper(indent + " ")*
  - If *hasRight() res += right.ToStringHelper(indent + " ")*
  - One string is built up, containing new lines
  - Return *res*

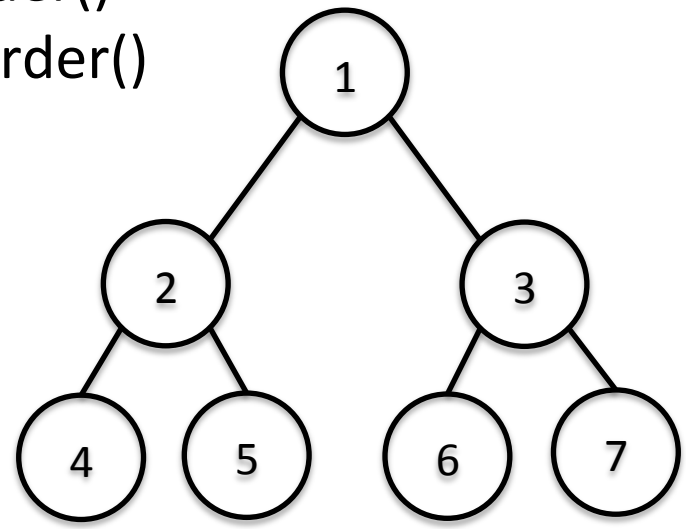
Agenda

1. General-purpose binary trees
2. Tree traversal
3. Expression trees

There are different ways to traverse a tree, depending on what needs to be done

preorder()

do something
left.preorder()
right.preorder()



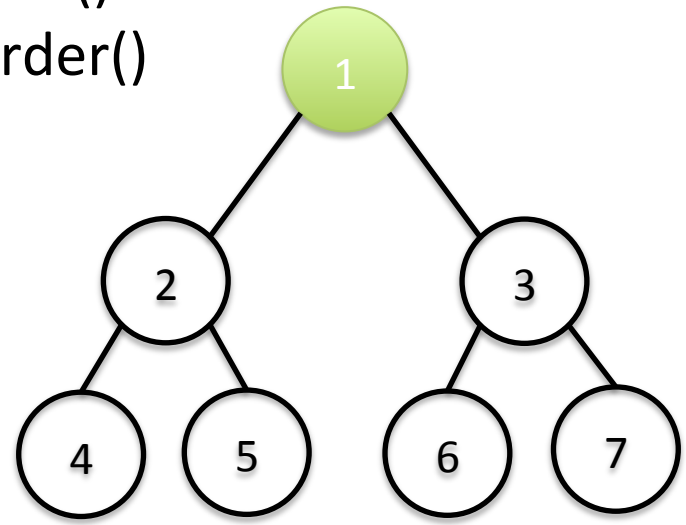
Examples:

File directory structure
Table of contents in book

There are different ways to traverse a tree, depending on what needs to be done

preorder()

do something
left.preorder()
right.preorder()



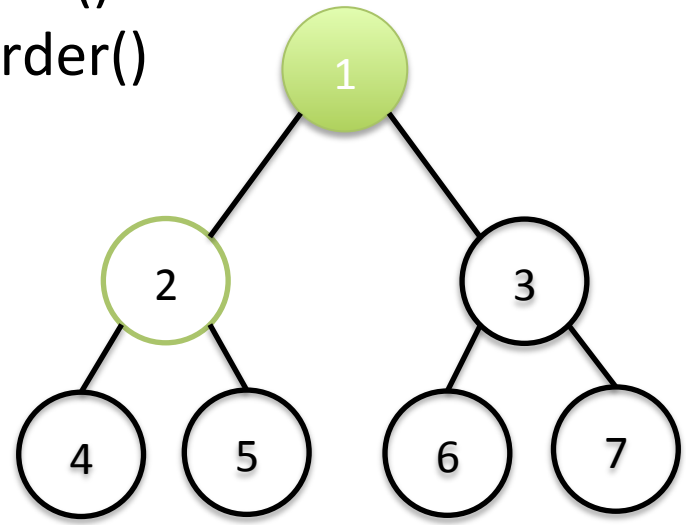
Visited
1

Examples:
File directory structure
Table of contents in book

There are different ways to traverse a tree, depending on what needs to be done

preorder()

do something
left.preorder()
right.preorder()



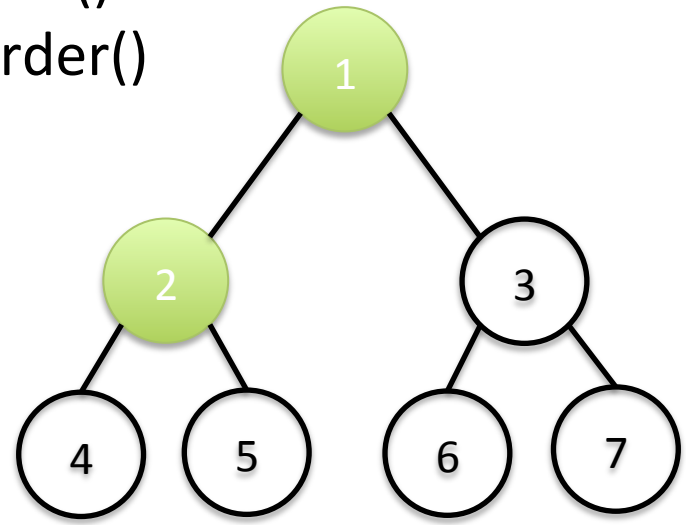
Visited
1

Examples:
File directory structure
Table of contents in book

There are different ways to traverse a tree, depending on what needs to be done

preorder()

do something
left.preorder()
right.preorder()



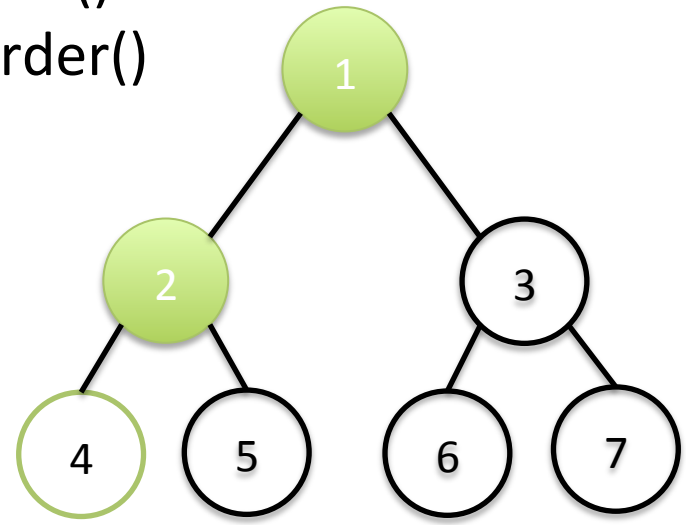
Visited
1, 2

Examples:
File directory structure
Table of contents in book

There are different ways to traverse a tree, depending on what needs to be done

preorder()

do something
left.preorder()
right.preorder()



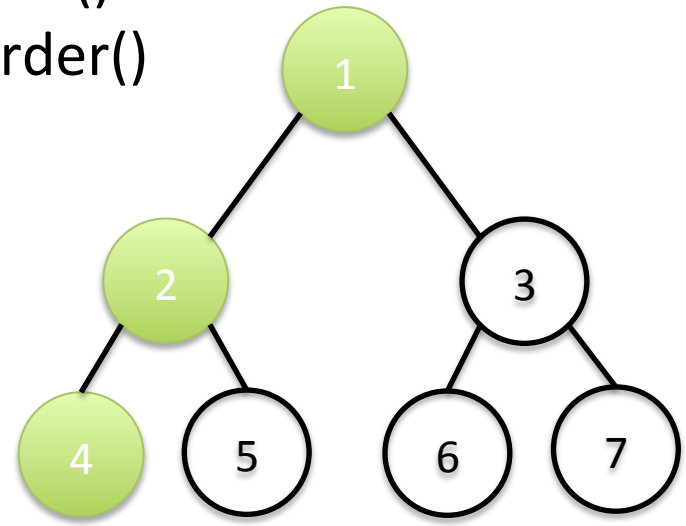
Visited
1, 2

Examples:
File directory structure
Table of contents in book

There are different ways to traverse a tree, depending on what needs to be done

preorder()

do something
left.preorder()
right.preorder()



Visited

1, 2, 4

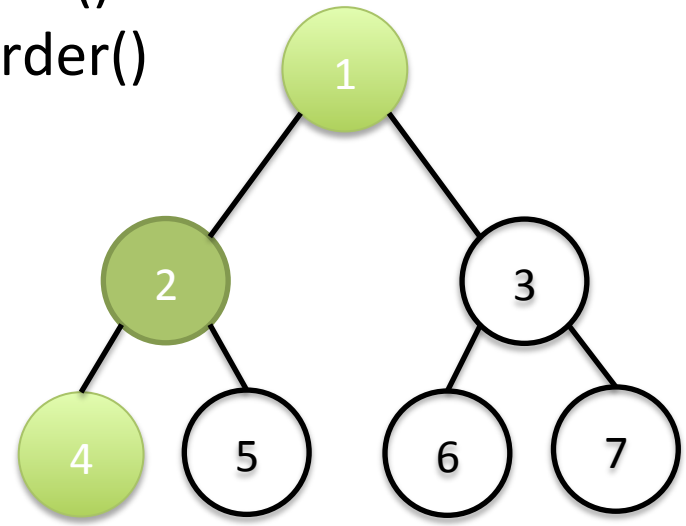
Examples:

File directory structure
Table of contents in book

There are different ways to traverse a tree, depending on what needs to be done

preorder()

do something
left.preorder()
right.preorder()



Visited

1, 2, 4

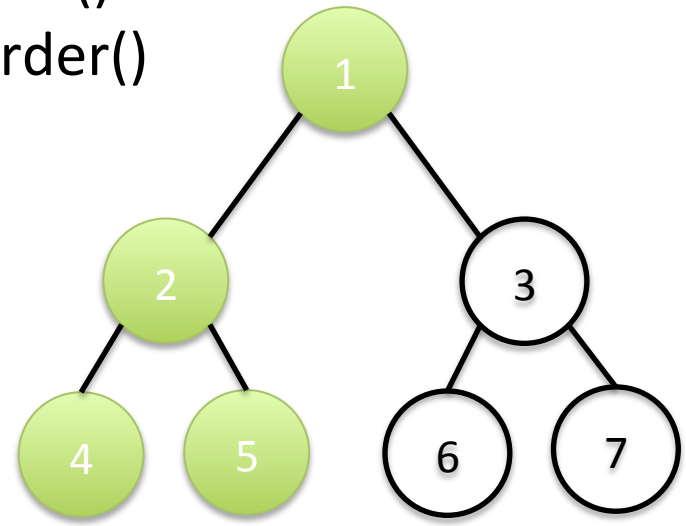
Examples:

File directory structure
Table of contents in book

There are different ways to traverse a tree, depending on what needs to be done

preorder()

do something
left.preorder()
right.preorder()



Visited

1, 2, 4, 5

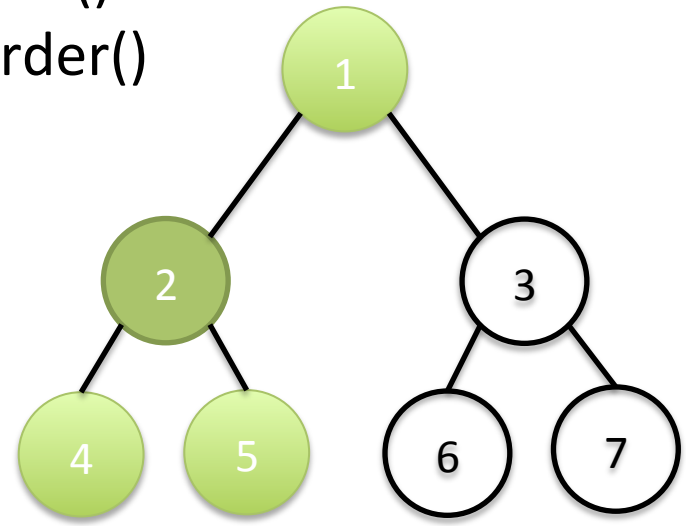
Examples:

File directory structure
Table of contents in book

There are different ways to traverse a tree, depending on what needs to be done

preorder()

do something
left.preorder()
right.preorder()



Visited

1, 2, 4, 5

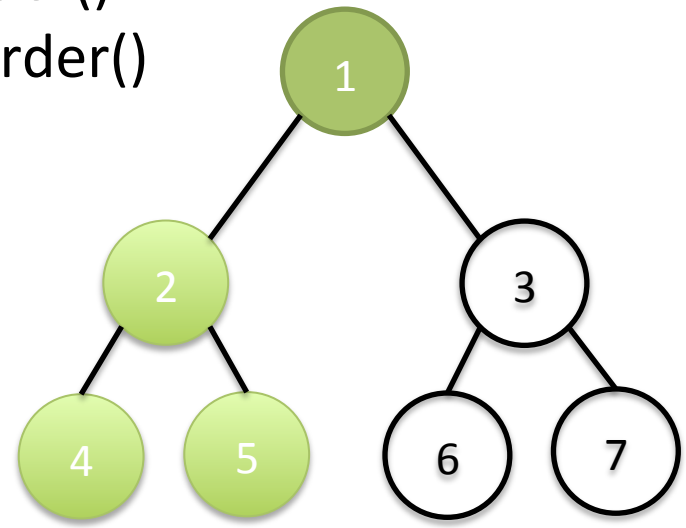
Examples:

File directory structure
Table of contents in book

There are different ways to traverse a tree, depending on what needs to be done

preorder()

do something
left.preorder()
right.preorder()



Visited

1, 2, 4, 5

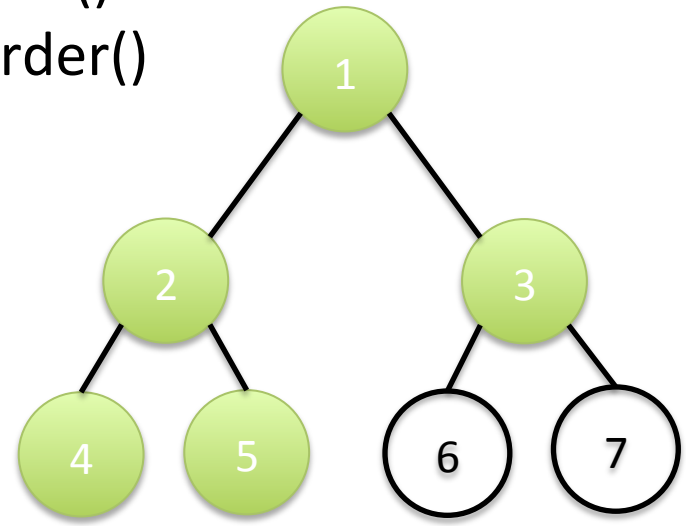
Examples:

File directory structure
Table of contents in book

There are different ways to traverse a tree, depending on what needs to be done

preorder()

do something
left.preorder()
right.preorder()



Visited

1, 2, 4, 5, 3

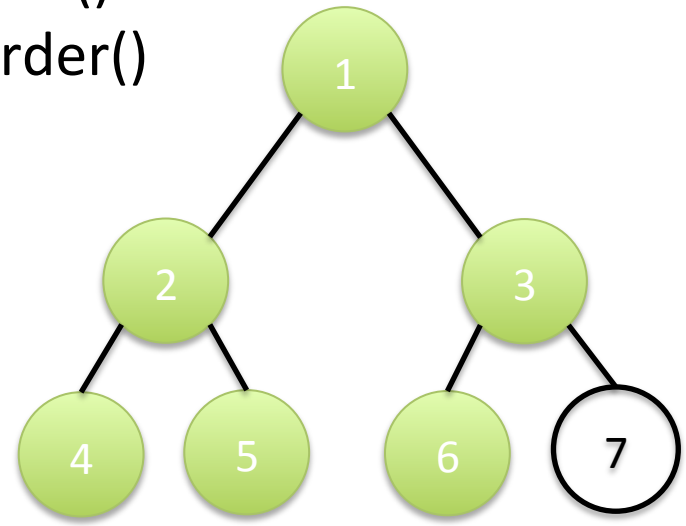
Examples:

File directory structure
Table of contents in book

There are different ways to traverse a tree, depending on what needs to be done

preorder()

do something
left.preorder()
right.preorder()



Visited

1, 2, 4, 5, 3, 6

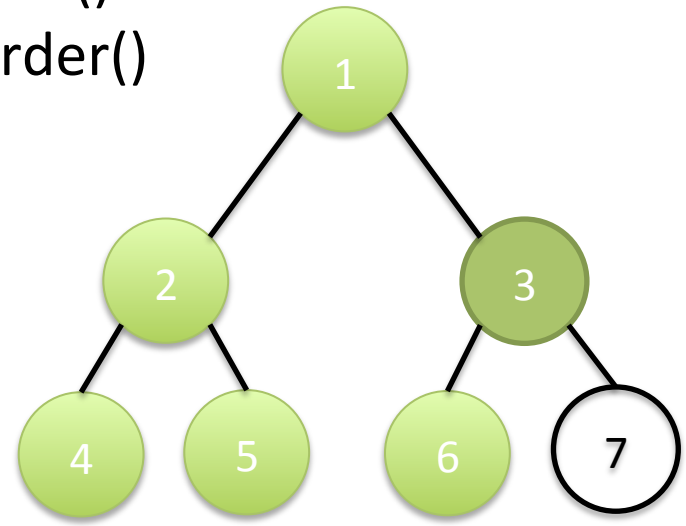
Examples:

File directory structure
Table of contents in book

There are different ways to traverse a tree, depending on what needs to be done

preorder()

do something
left.preorder()
right.preorder()



Visited

1, 2, 4, 5, 3, 6

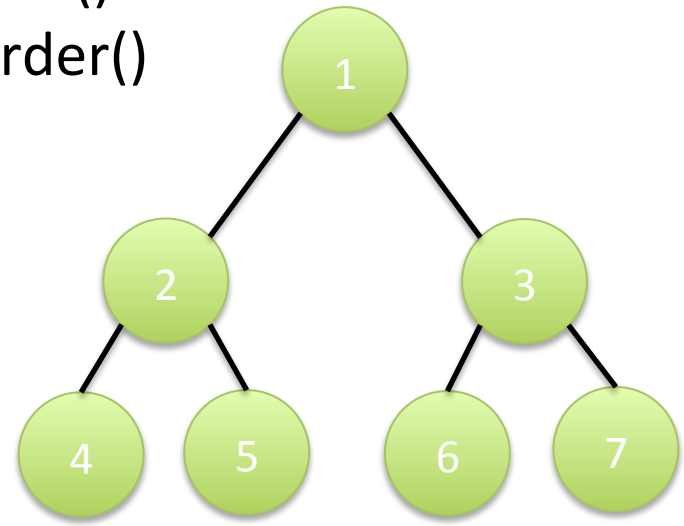
Examples:

File directory structure
Table of contents in book

There are different ways to traverse a tree, depending on what needs to be done

preorder()

do something
left.preorder()
right.preorder()



Visited

1, 2, 4, 5, 3, 6, 7

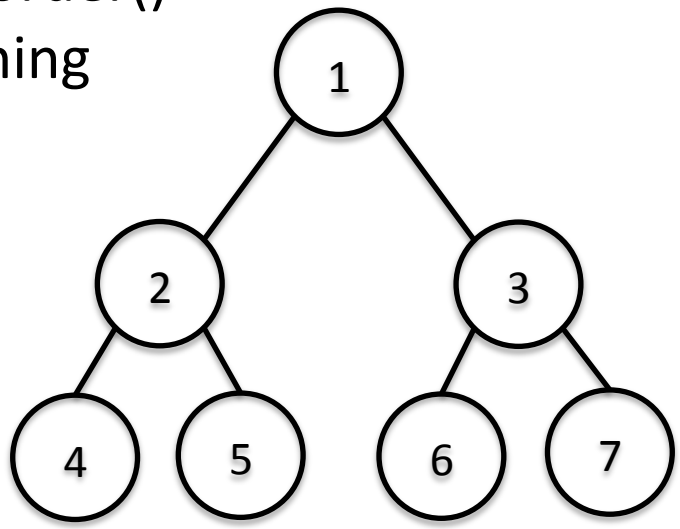
Examples:

File directory structure
Table of contents in book

There are different ways to traverse a tree, depending on what needs to be done

postorder()

left.postorder()
right.postorder()
do something



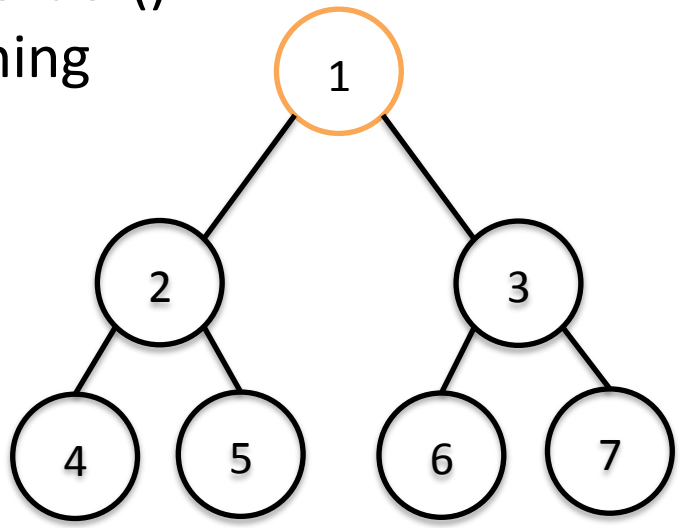
Example:

Compute disk space (not sure how many bytes in each directory until you search all children)

There are different ways to traverse a tree, depending on what needs to be done

postorder()

left.postorder()
right.postorder()
do something



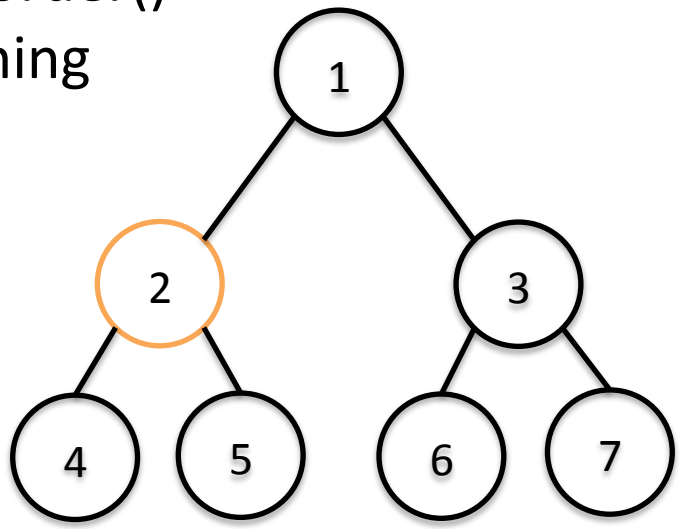
Example:

Compute disk space (not sure how many bytes in each directory until you search all children)

There are different ways to traverse a tree, depending on what needs to be done

postorder()

left.postorder()
right.postorder()
do something



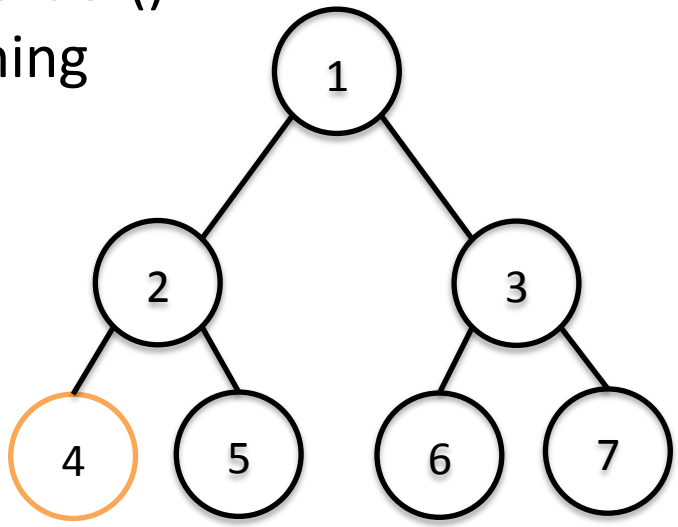
Example:

Compute disk space (not sure how many bytes in each directory until you search all children)

There are different ways to traverse a tree, depending on what needs to be done

postorder()

left.postorder()
right.postorder()
do something



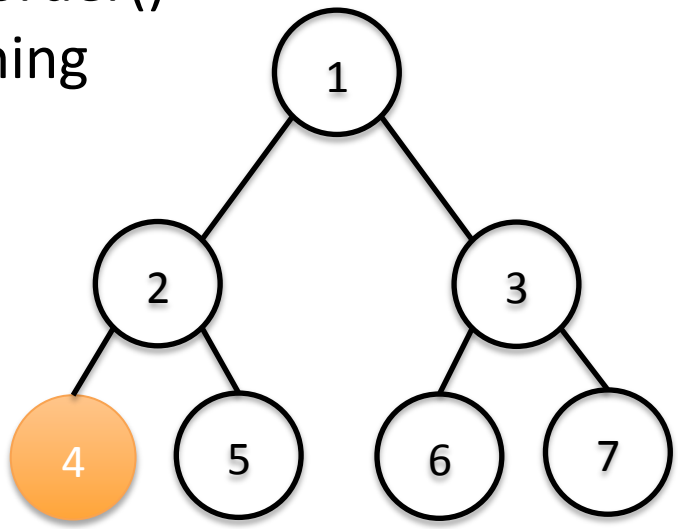
Example:

Compute disk space (not sure how many bytes in each directory until you search all children)

There are different ways to traverse a tree, depending on what needs to be done

postorder()

left.postorder()
right.postorder()
do something



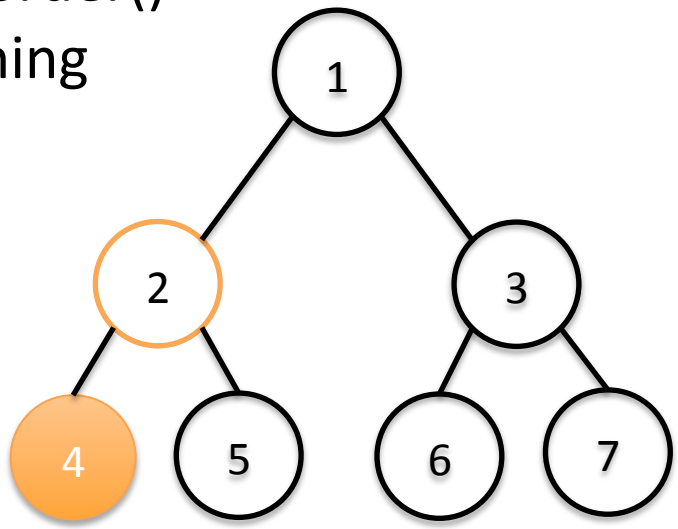
Visited
4

Example:
Compute disk space (not sure how many bytes in each directory until you search all children)

There are different ways to traverse a tree, depending on what needs to be done

postorder()

left.postorder()
right.postorder()
do something



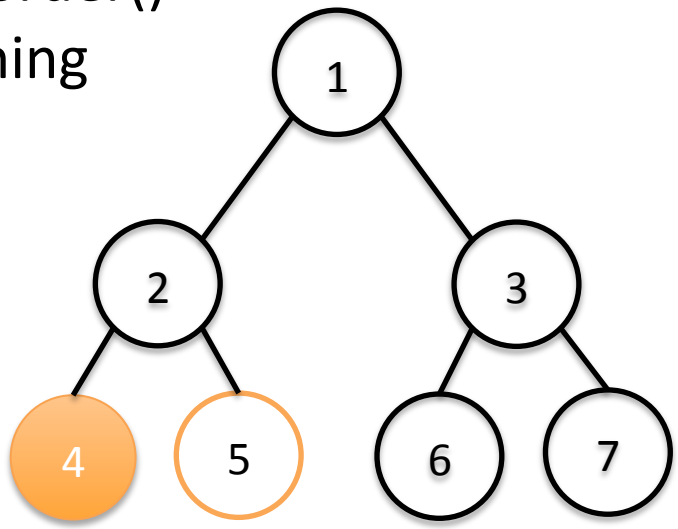
Visited
4

Example:
Compute disk space (not sure how many bytes in each directory until you search all children)

There are different ways to traverse a tree, depending on what needs to be done

postorder()

left.postorder()
right.postorder()
do something



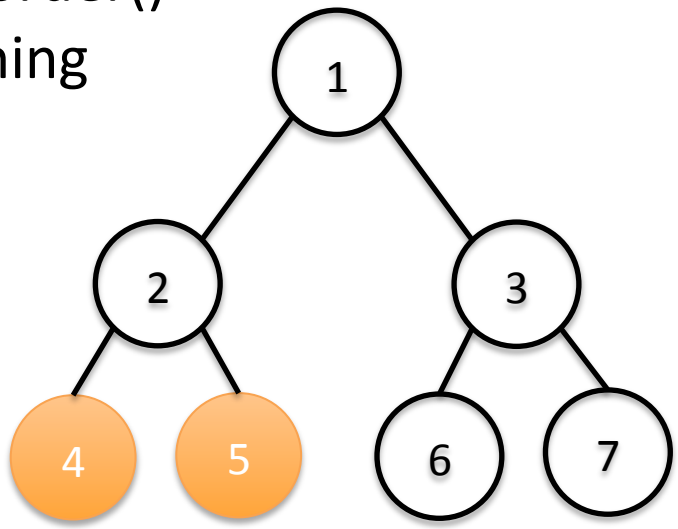
Visited
4

Example:
Compute disk space (not sure how many bytes in each directory until you search all children)

There are different ways to traverse a tree, depending on what needs to be done

postorder()

left.postorder()
right.postorder()
do something



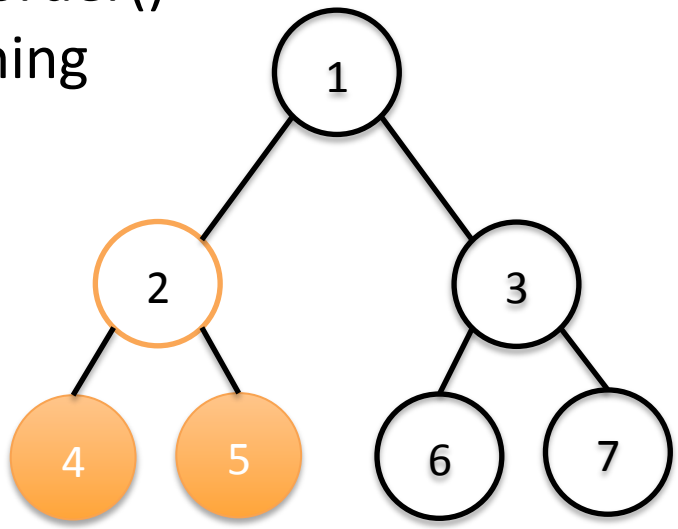
Visited
4, 5

Example:
Compute disk space (not sure how many bytes in each directory until you search all children)

There are different ways to traverse a tree, depending on what needs to be done

postorder()

left.postorder()
right.postorder()
do something



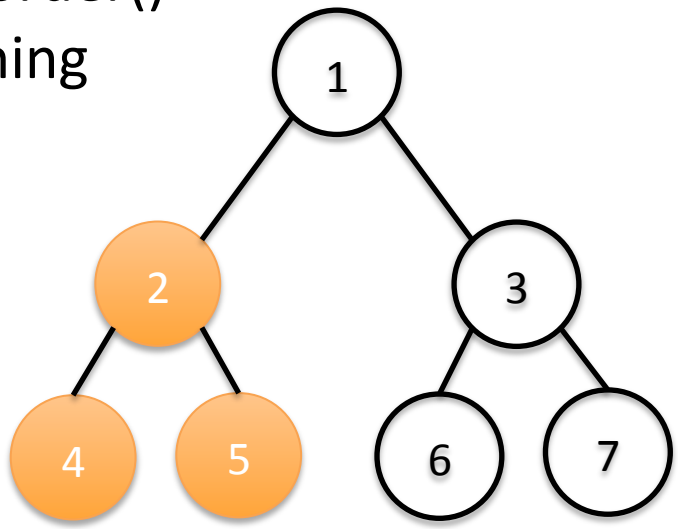
Visited
4, 5

Example:
Compute disk space (not sure how many bytes in each directory until you search all children)

There are different ways to traverse a tree, depending on what needs to be done

postorder()

left.postorder()
right.postorder()
do something



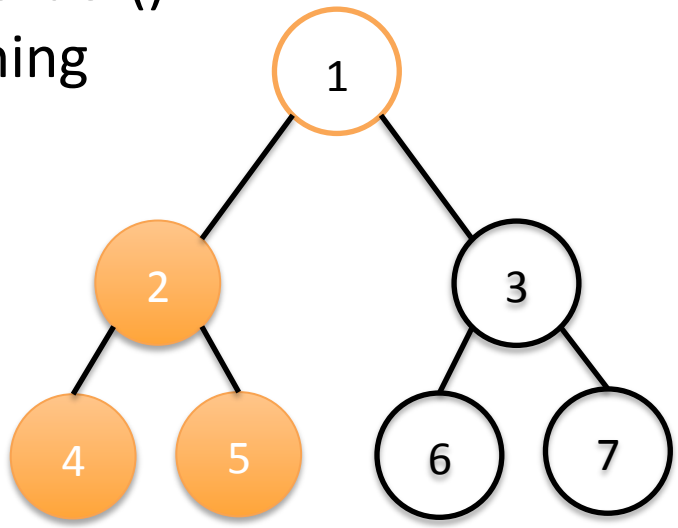
Visited
4, 5, 2

Example:
Compute disk space (not sure how many bytes in each directory until you search all children)

There are different ways to traverse a tree, depending on what needs to be done

postorder()

left.postorder()
right.postorder()
do something



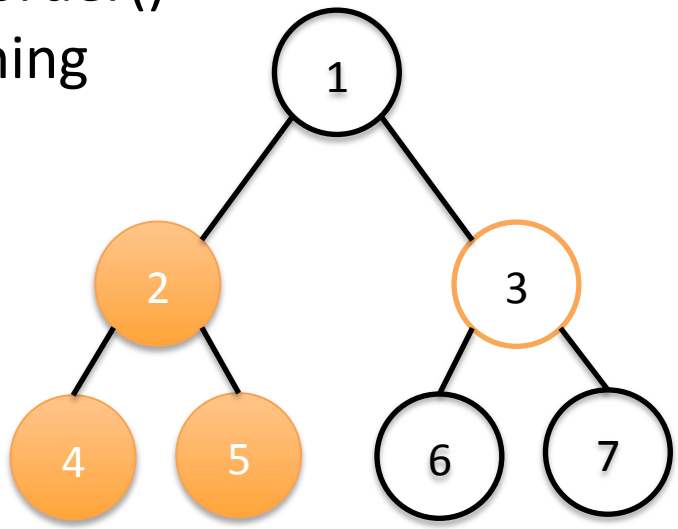
Visited
4, 5, 2

Example:
Compute disk space (not sure how many bytes in each directory until you search all children)

There are different ways to traverse a tree, depending on what needs to be done

postorder()

left.postorder()
right.postorder()
do something



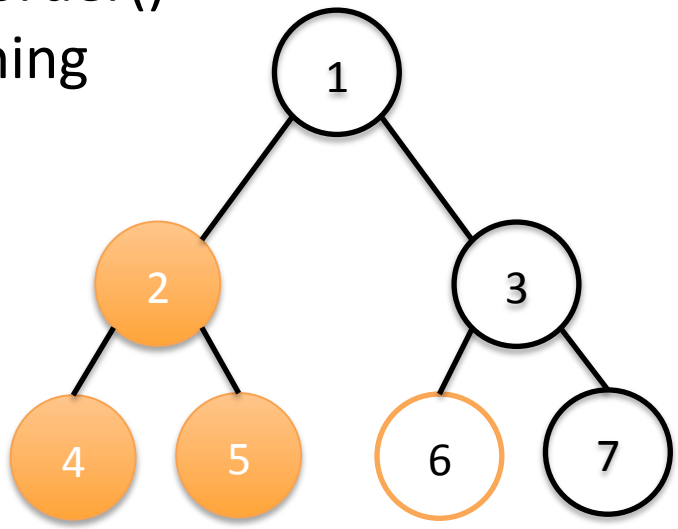
Visited
4, 5, 2

Example:
Compute disk space (not sure how many bytes in each directory until you search all children)

There are different ways to traverse a tree, depending on what needs to be done

postorder()

left.postorder()
right.postorder()
do something



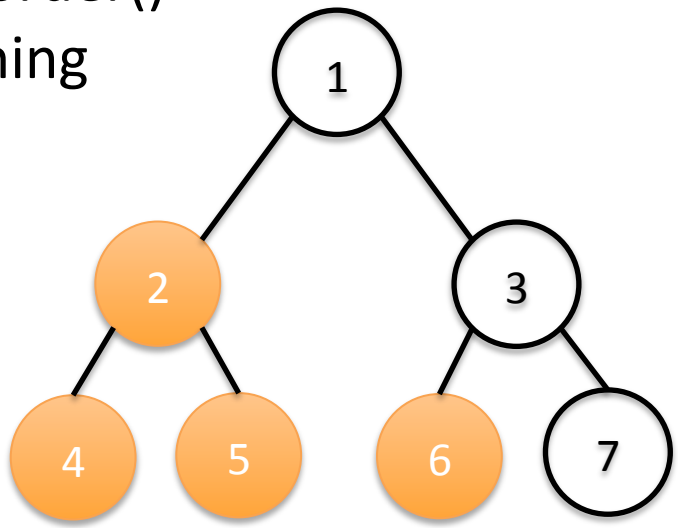
Visited
4, 5, 2

Example:
Compute disk space (not sure how many bytes in each directory until you search all children)

There are different ways to traverse a tree, depending on what needs to be done

postorder()

left.postorder()
right.postorder()
do something



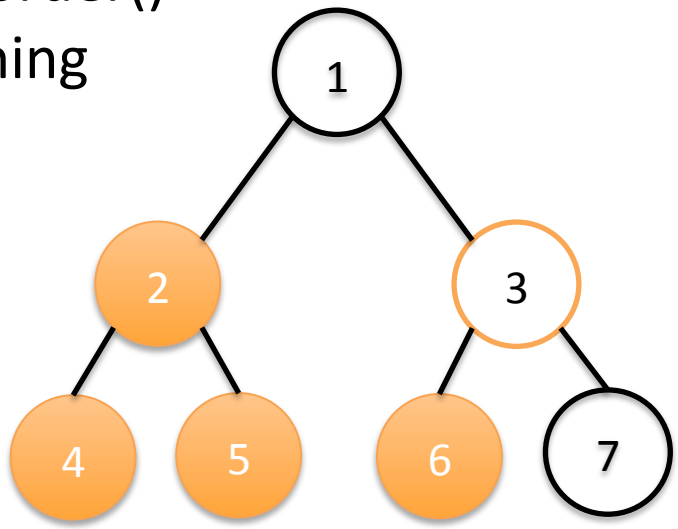
Visited
4, 5, 2, 6

Example:
Compute disk space (not sure how many bytes in each directory until you search all children)

There are different ways to traverse a tree, depending on what needs to be done

postorder()

left.postorder()
right.postorder()
do something



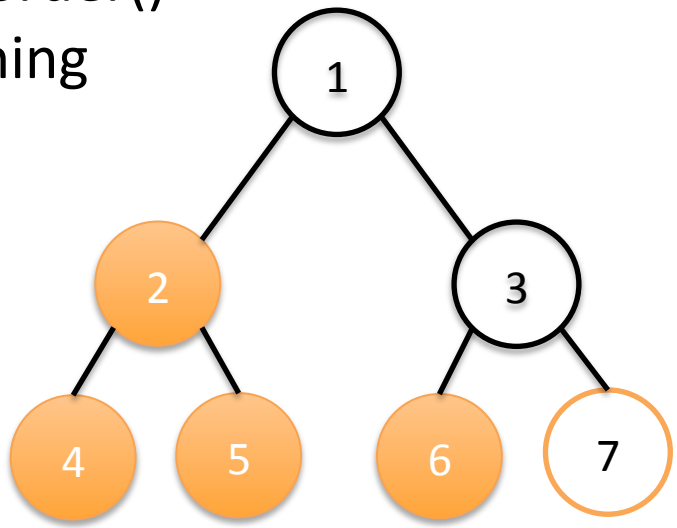
Visited
4, 5, 2, 6

Example:
Compute disk space (not sure how many bytes in each directory until you search all children)

There are different ways to traverse a tree, depending on what needs to be done

postorder()

left.postorder()
right.postorder()
do something



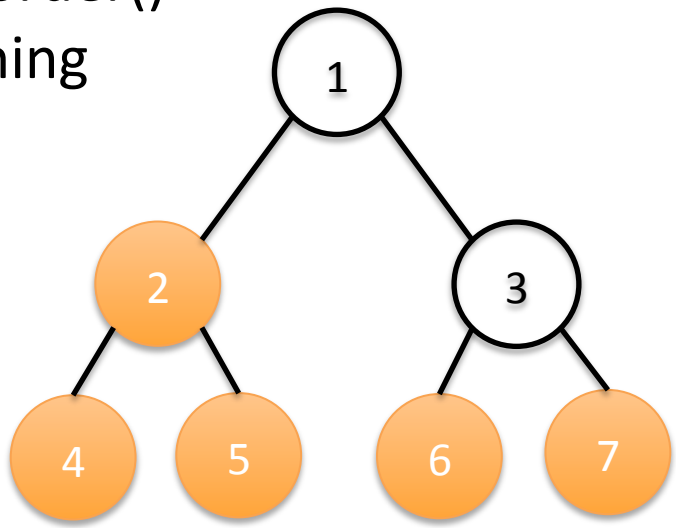
Visited
4, 5, 2, 6

Example:
Compute disk space (not sure how many bytes in each directory until you search all children)

There are different ways to traverse a tree, depending on what needs to be done

postorder()

left.postorder()
right.postorder()
do something



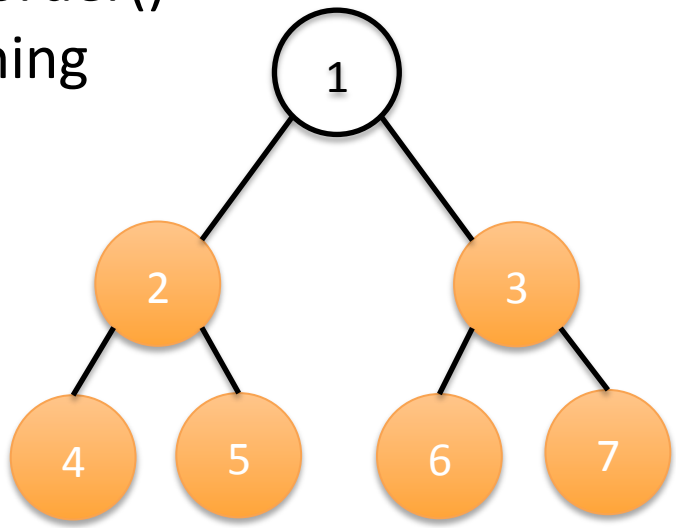
Visited
4, 5, 2, 6, 7

Example:
Compute disk space (not sure how many bytes in each directory until you search all children)

There are different ways to traverse a tree, depending on what needs to be done

postorder()

left.postorder()
right.postorder()
do something



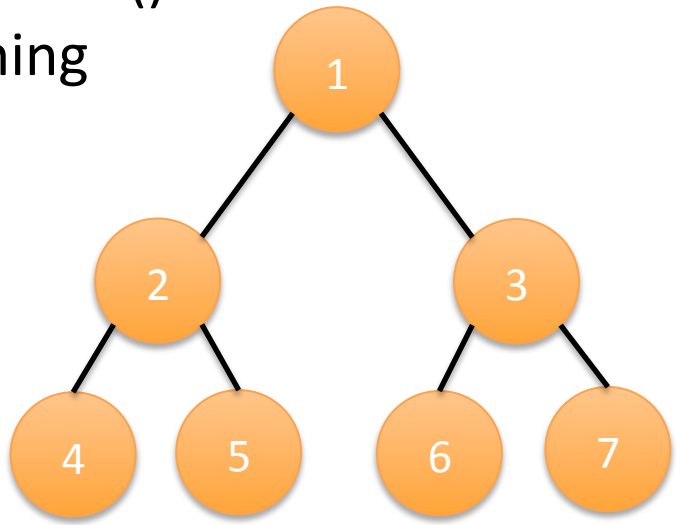
Visited
4, 5, 2, 6, 7, 3

Example:
Compute disk space (not sure how many bytes in each directory until you search all children)

There are different ways to traverse a tree, depending on what needs to be done

postorder()

left.postorder()
right.postorder()
do something



Visited

4, 5, 2, 6, 7, 3, 1

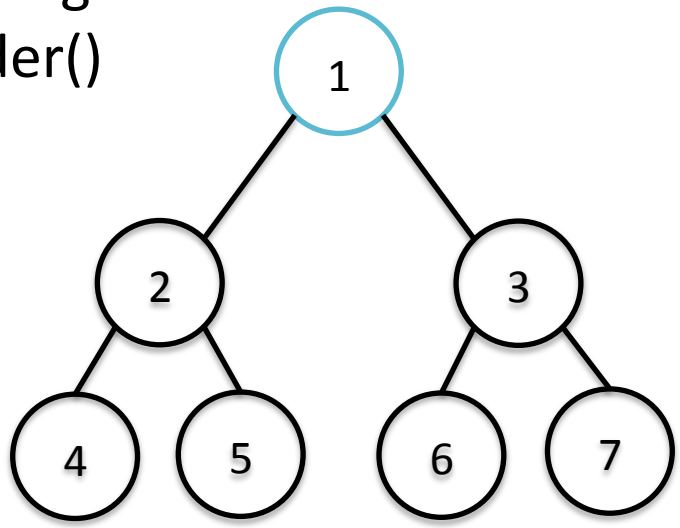
Example:

Compute disk space (not sure how many bytes in each directory until you search all children)

There are different ways to traverse a tree, depending on what needs to be done

inorder()

left.inorder()
do something
right.inorder()



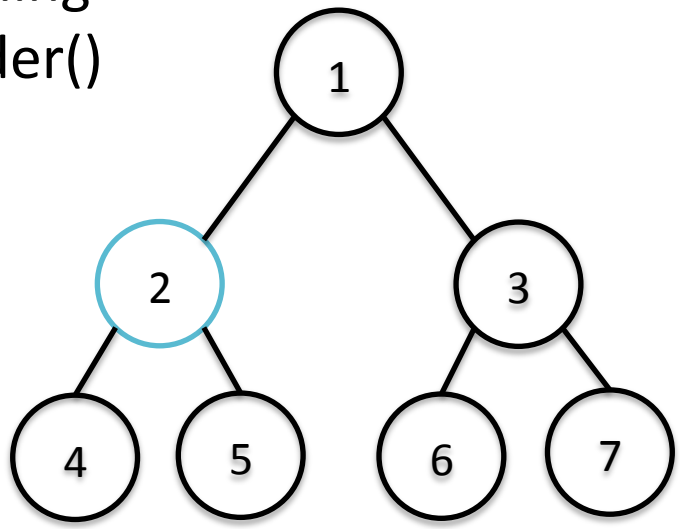
Example:

Drawing a tree

There are different ways to traverse a tree, depending on what needs to be done

inorder()

left.inorder()
do something
right.inorder()



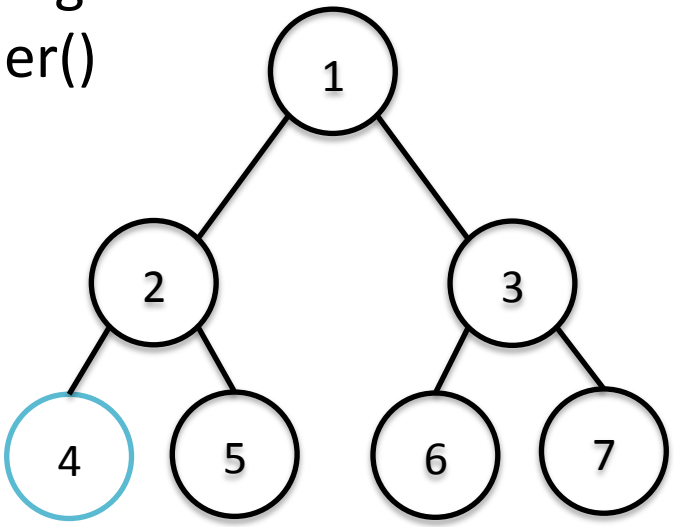
Example:

Drawing a tree

There are different ways to traverse a tree, depending on what needs to be done

inorder()

left.inorder()
do something
right.inorder()



Example:

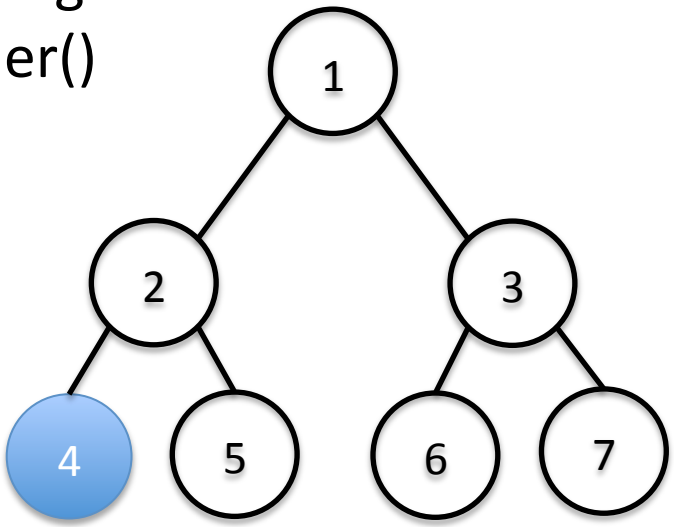
Drawing a tree

There are different ways to traverse a tree, depending on what needs to be done

inorder()

left.inorder()
do something
right.inorder()

Example:
Drawing a tree



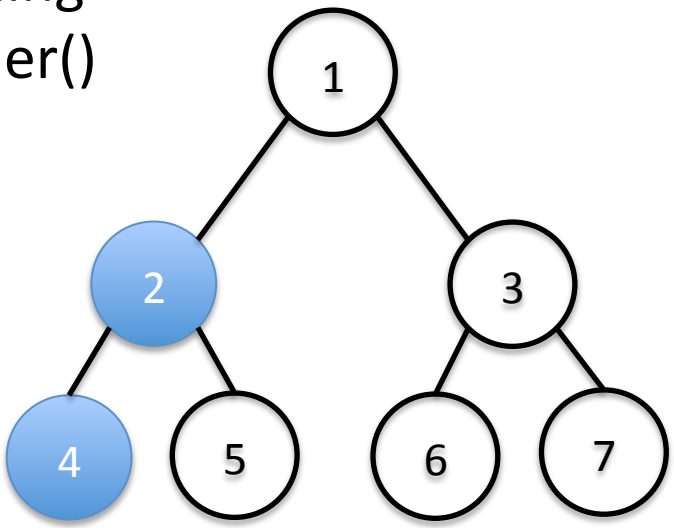
Visited
4

There are different ways to traverse a tree, depending on what needs to be done

inorder()

left.inorder()
do something
right.inorder()

Example:
Drawing a tree



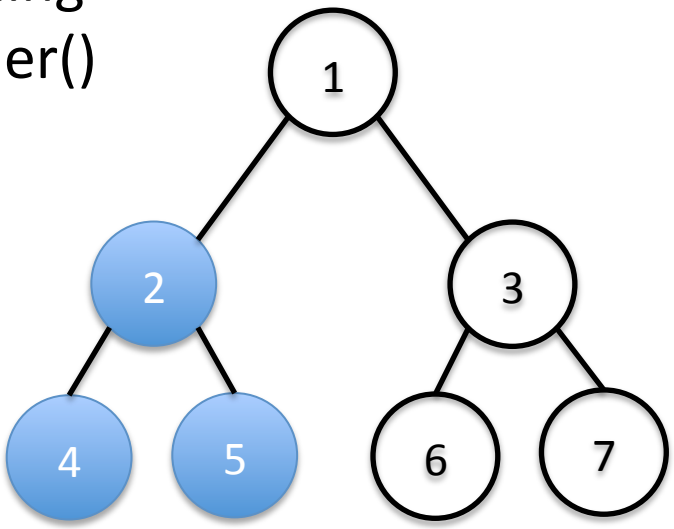
Visited
4, 2

There are different ways to traverse a tree, depending on what needs to be done

inorder()

left.inorder()
do something
right.inorder()

Example:
Drawing a tree



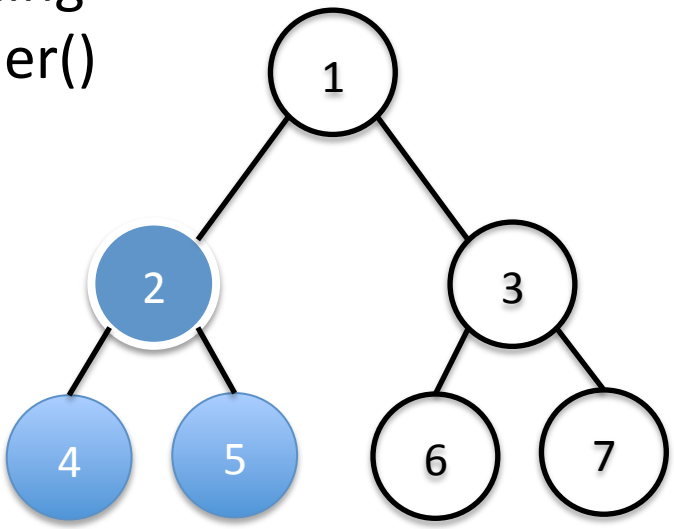
Visited
4, 2, 5

There are different ways to traverse a tree, depending on what needs to be done

inorder()

left.inorder()
do something
right.inorder()

Example:
Drawing a tree



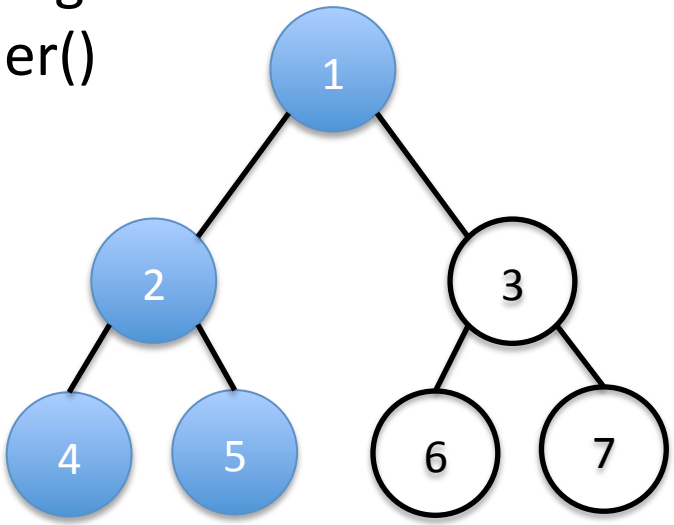
Visited
4, 2, 5

There are different ways to traverse a tree, depending on what needs to be done

inorder()

left.inorder()
do something
right.inorder()

Example:
Drawing a tree



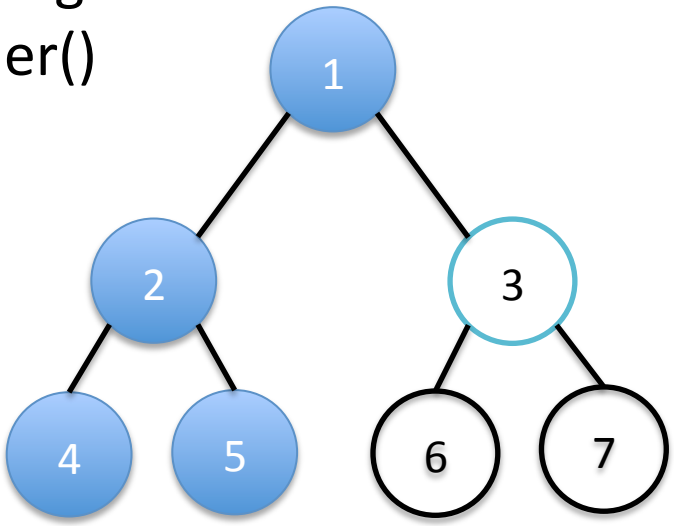
Visited
4, 2, 5, 1

There are different ways to traverse a tree, depending on what needs to be done

inorder()

left.inorder()
do something
right.inorder()

Example:
Drawing a tree



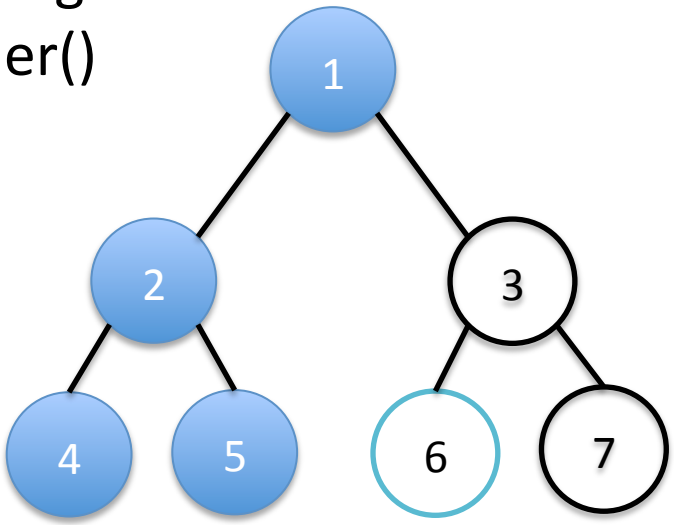
Visited
4, 2, 5, 1

There are different ways to traverse a tree, depending on what needs to be done

inorder()

left.inorder()
do something
right.inorder()

Example:
Drawing a tree



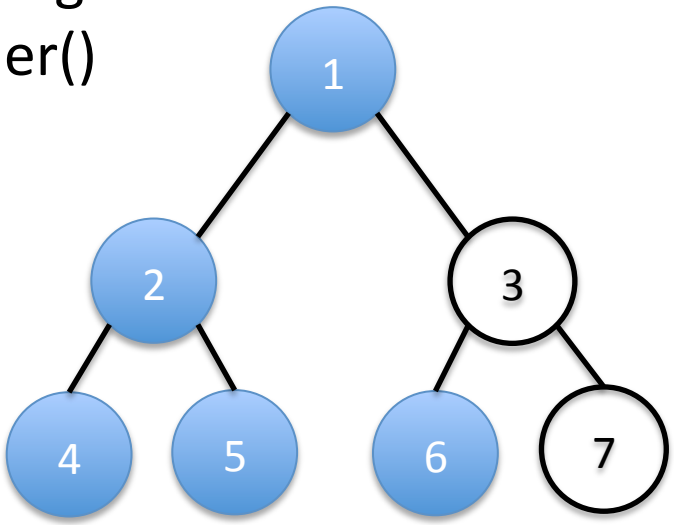
Visited
4, 2, 5, 1

There are different ways to traverse a tree, depending on what needs to be done

inorder()

left.inorder()
do something
right.inorder()

Example:
Drawing a tree



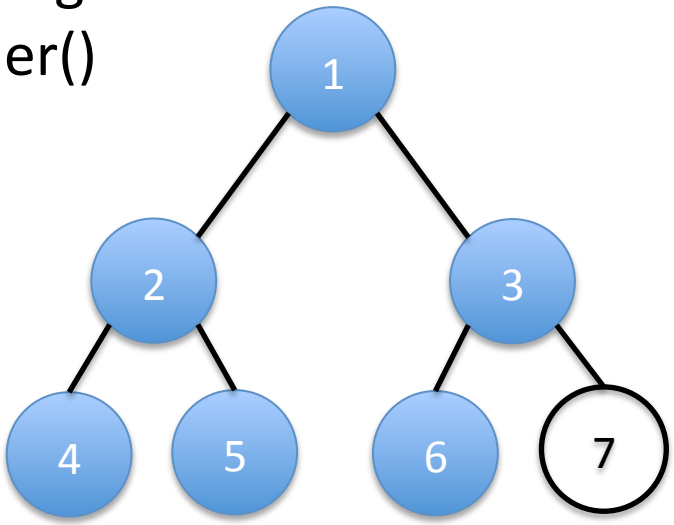
Visited
4, 2, 5, 1, 6

There are different ways to traverse a tree, depending on what needs to be done

inorder()

left.inorder()
do something
right.inorder()

Example:
Drawing a tree



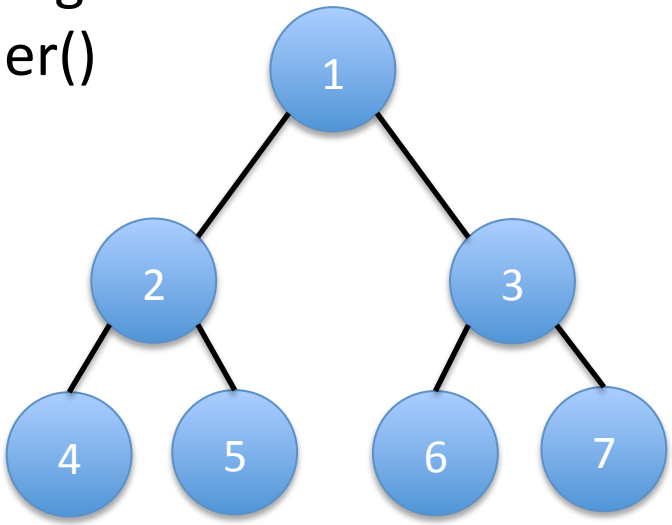
Visited
4, 2, 5, 1, 6, 3

There are different ways to traverse a tree, depending on what needs to be done

inorder()

left.inorder()
do something
right.inorder()

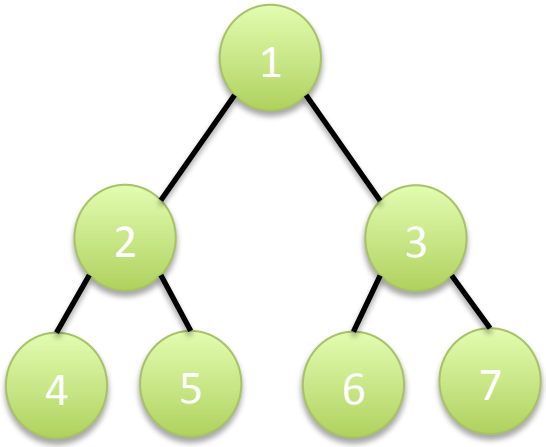
Example:
Drawing a tree



Visited
4, 2, 5, 1, 6, 3, 7

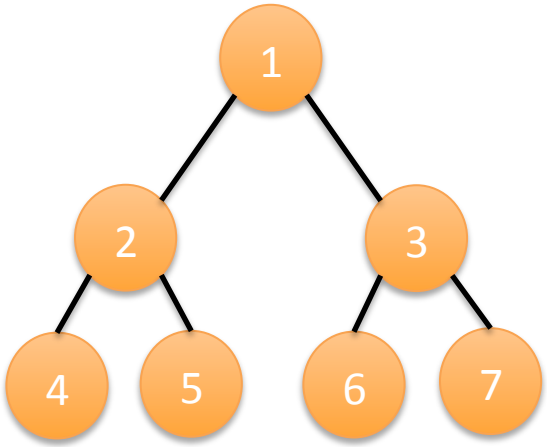
Summary

Preorder



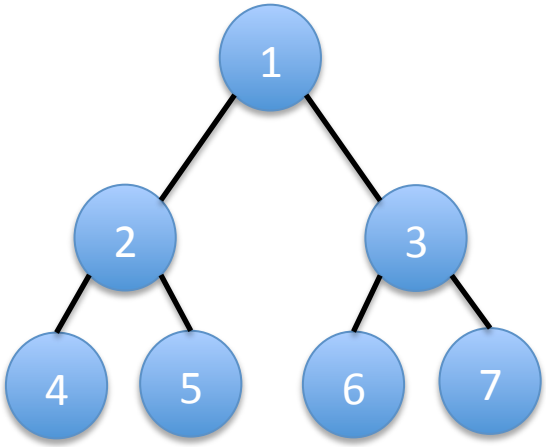
Visited
1, 2, 4, 5, 3, 6, 7
Book chapters

Postorder



Visited
4, 5, 2, 6, 7, 3, 1
Calculate disk space

Inorder



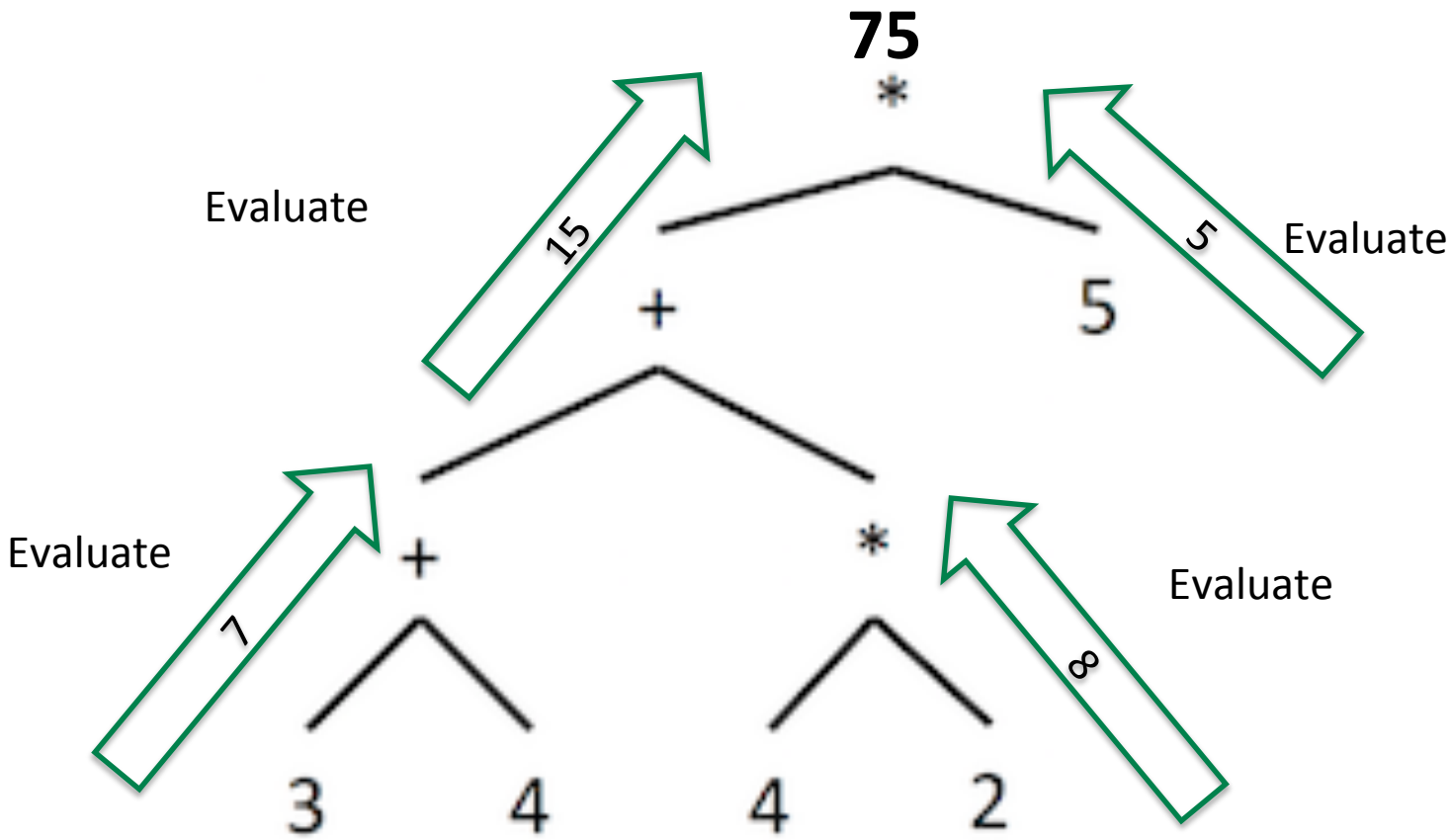
Visited
4, 2, 5, 1, 6, 3, 7
Drawing a tree
(left to right)

Agenda

1. General-purpose binary trees
2. Tree traversal
3. Expression trees

Expression trees ask their children to evaluate themselves

Expression tree for ((3+4) + (4*2)) * 5



Expression Tree builds Expression nodes and evaluates them

Expression.java

- Interface, specifying that nodes must provide evaluate method

Number.java

- Implements Expression interface
- Holds double
- Returns that value on *evaluate()*

Addition.java

- Implements Expression interface
- Holds left and right Expression
- Returns left + right on *evaluate()*

Multiplication

- Implements Expression interface
- Holds left and right Expression
- Returns left * right on *evaluate()*

Expression Tree builds Expression nodes and evaluates them

ExpressionTest.java

- Creates Number and evaluates
- Creates 3 + 4 calling new Addition and passing two Numbers
- Creates test by calling helper function below (for readability)
  - *c()* creates new Number
  - *plus (e1,e2)* creates new Addition
  - *times(e1,e2)* creates new Multiplication

- Run