CSC 148: Introduction to Computer Science

Week 9

Tree traversals

Binary Search Trees (BSTs)

University of Toronto Mississauga,

Department of Mathematical and Computational Sciences

Today: Tree Traversals

- Formalize strategies for traversing any general Tree
  - BSTs are just a special type of Tree …
  - Hence, traversals are similar

Functions as Arguments

```
def list_if(self, p: Callable[[Any], bool]) -> list[Any]:
    """Return a list of values in this Tree that satisfy predicate <p>.

    Parameter <p> is a function which takes a single value argument and
    checks some predicate on that value, returning True or False.
    """
    pass
```

Idea: if self._root is empty => return []

if it's a leaf => [self._root] if p(self._root) else []

otherwise => call recursively for each subtree in self._subtrees

and construct the list of values that satisfy p

Traversal

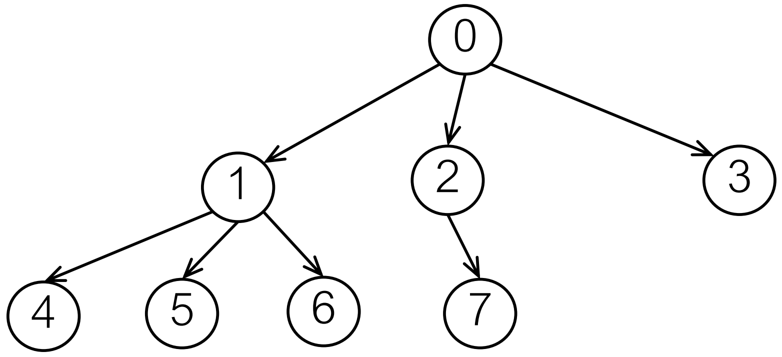
- The functions and methods we have seen get information from every node of the tree -- in some sense they traverse the tree

- Sometimes the order of processing tree nodes is important:
  - Do we process the root of the tree (and the root of each subtree...) before or after its children?
  - Or, perhaps, we process along levels that are the same distance from the root?

Preorder Traversal

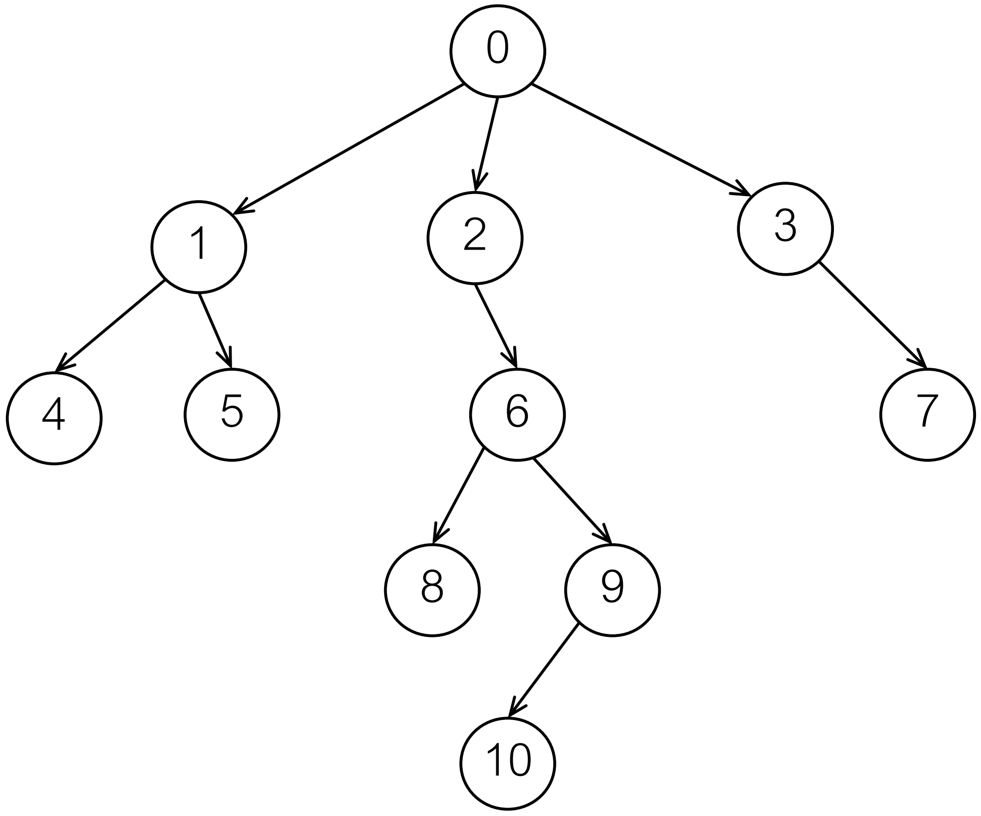
- Visit each node of Tree t as follows:
  - do something with the node / value, e.g., print it
  - visit its first subtree in preorder
  - visit its second subtree in preorder
  - ...

What is the sequence of nodes being visited in preorder?



Preorder – More Examples

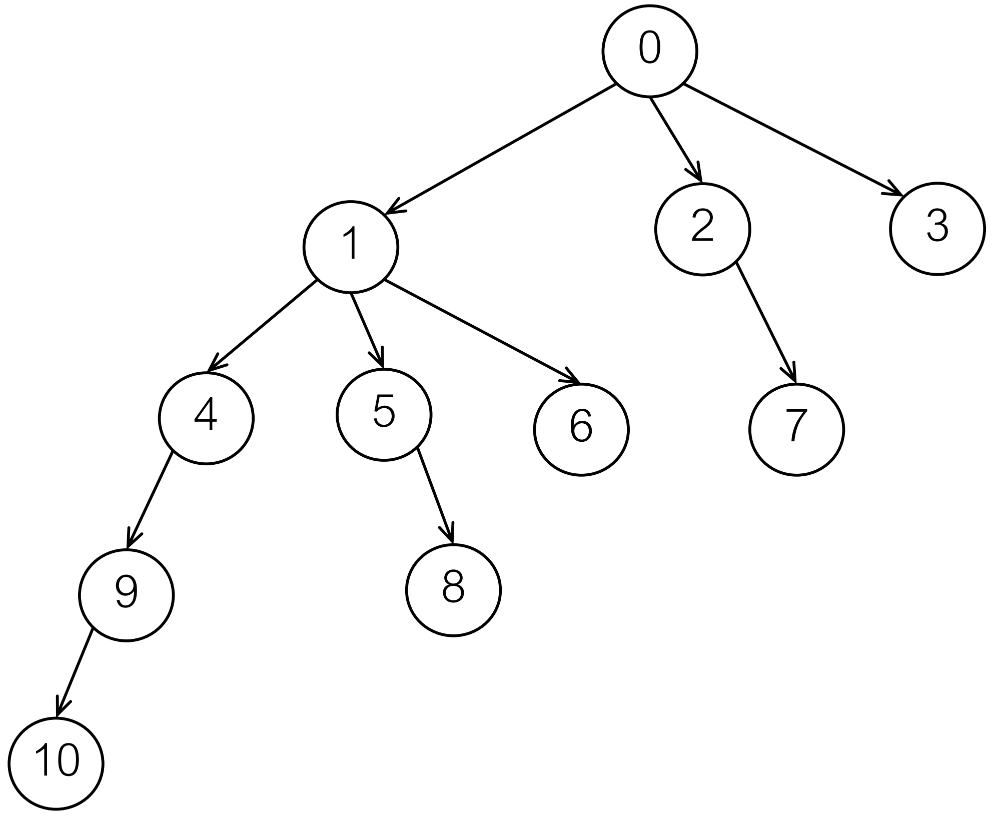
- Visit each node of Tree t as follows:
  - do something with the node / value, e.g., print it
  - visit its first subtree in preorder
  - visit its second subtree in preorder
  - ...



What is the sequence of nodes being visited in preorder?

Preorder - More Examples

- Visit each node of Tree t as follows:
  - do something with the node / value, e.g., print it
  - visit its first subtree in preorder
  - visit its second subtree in preorder
  - ...



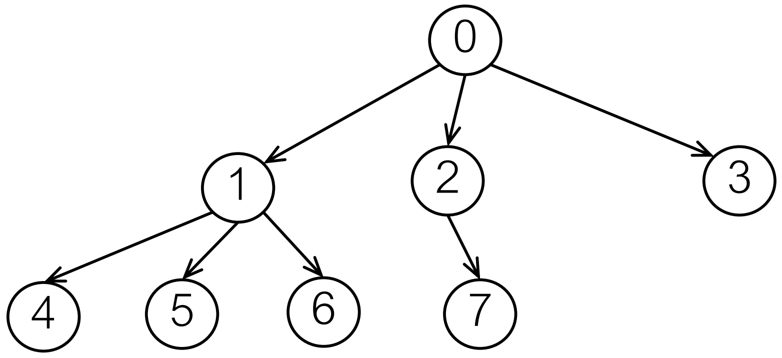
What is the sequence of nodes being visited in preorder?

Any thoughts on how to implement this visit order?

Postorder Traversal

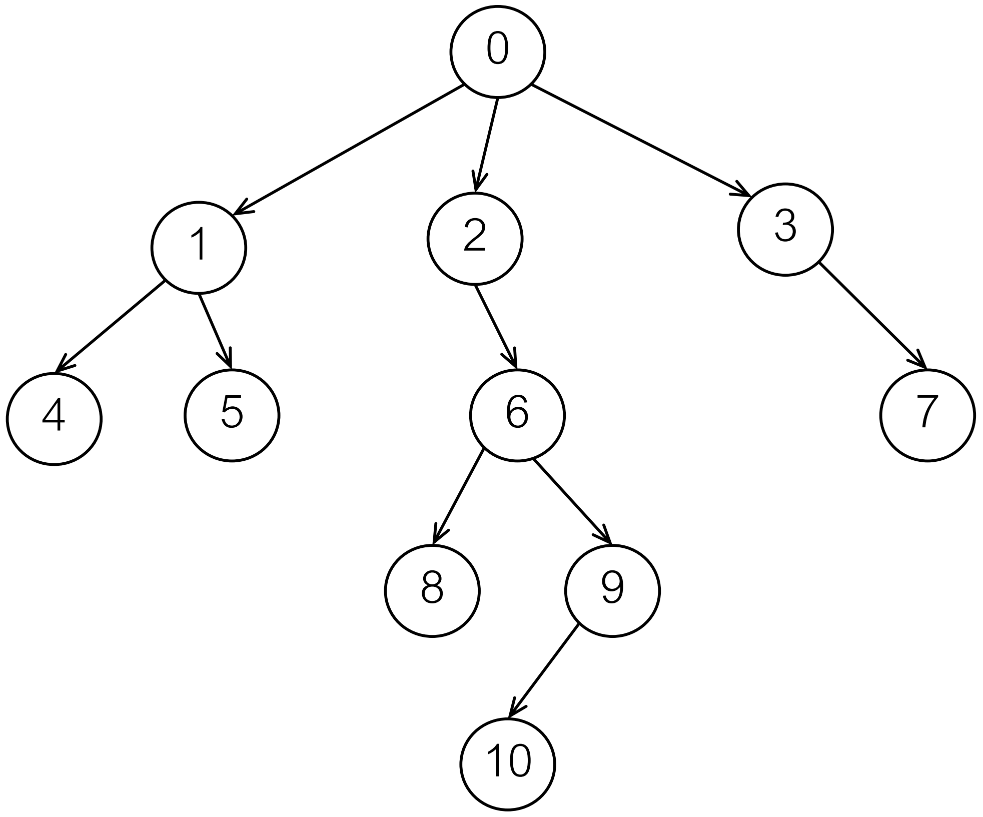
- Visit each node of Tree t as follows:
  - visit its first subtree in postorder
  - visit its second subtree in postorder
  - ...
  - do something with the node / value, e.g., print it

What is the sequence of nodes being visited in postorder?



Postorder – More Examples

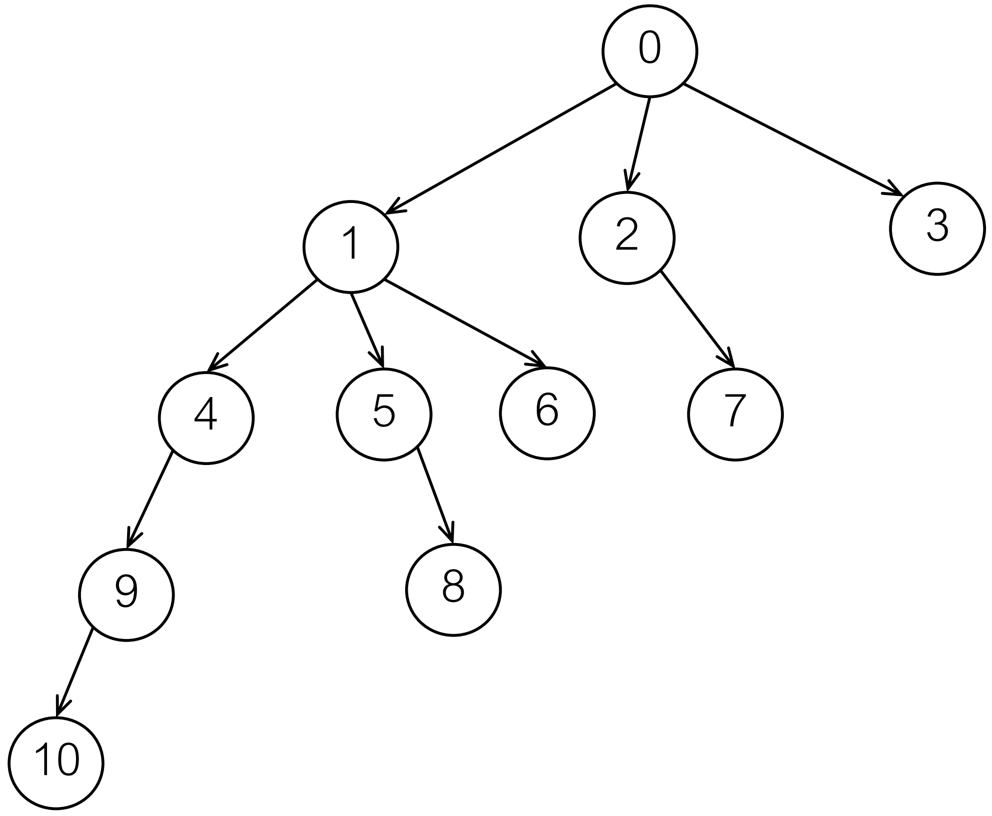
- Visit each node of Tree t as follows:
  - visit its first subtree in postorder
  - visit its second subtree in postorder
  - ...
  - do something with the node / value, e.g., print it



What is the sequence of nodes being visited in postorder?

Postorder – More Examples

- Visit each node of Tree t as follows:
  - visit its first subtree in postorder
  - visit its second subtree in postorder
  - ...
  - do something with the node / value, e.g., print it



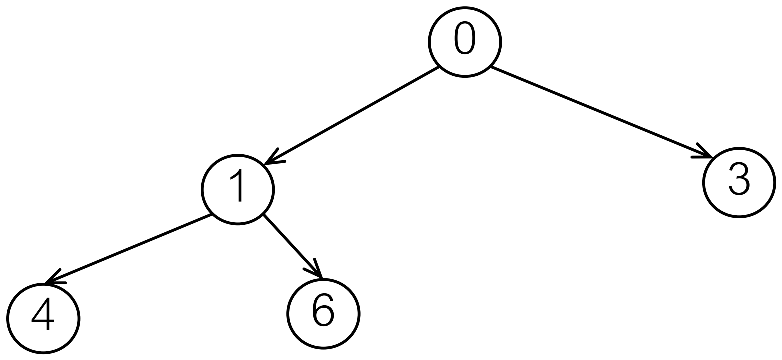
What is the sequence of nodes being visited in postorder?

Any thoughts on how to implement this visit order?

Inorder Traversal (BST only)

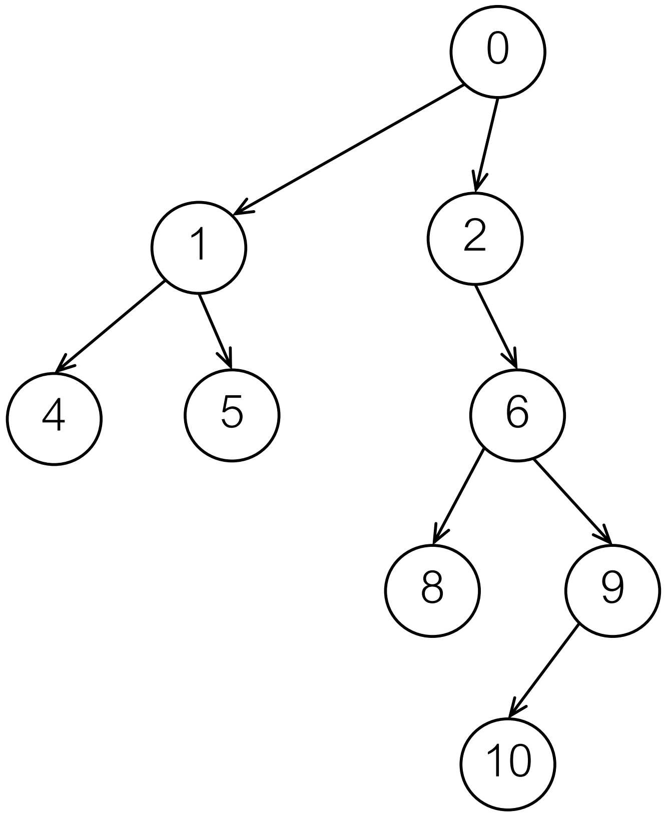
- Visit each node of Tree t as follows:
  - visit its first subtree in inorder
  - do something with the node / value, e.g., print it
  - visit its second subtree in inorder

What is the sequence of nodes being visited in inorder?



Inorder – More Examples

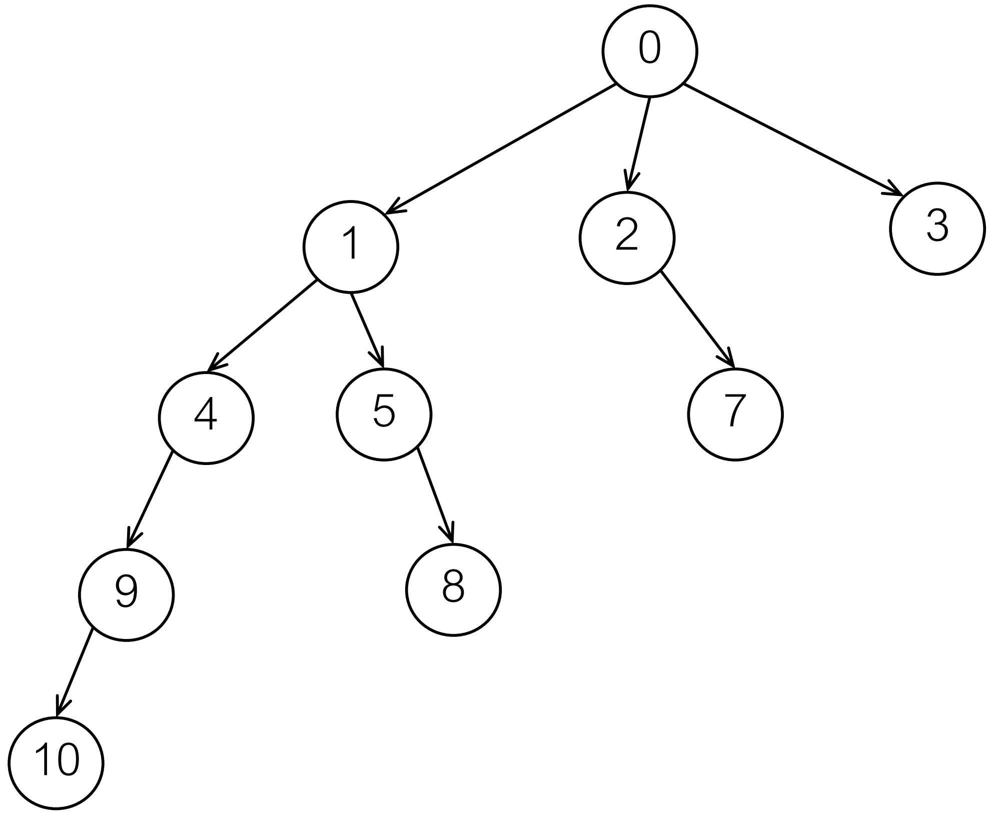
- Visit each node of Tree t as follows:
  - visit its first subtree in inorder
  - do something with the node / value, e.g., print it
  - visit its second subtree in inorder



What is the sequence of nodes being visited in inorder?

Inorder – More Examples

- Visit each node of Tree t as follows:
  - visit its first subtree in inorder
  - do something with the node / value, e.g., print it
  - visit its second subtree in inorder



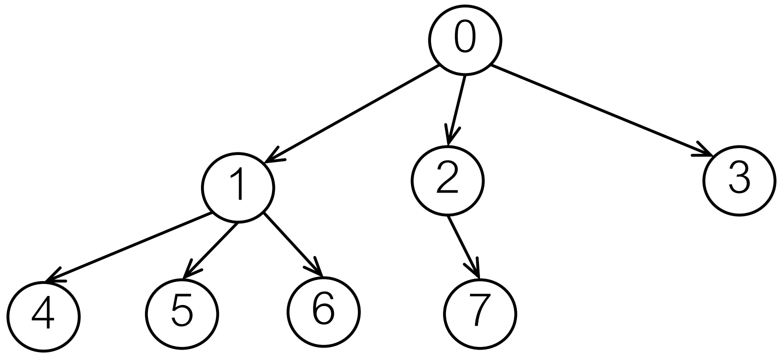
What is the sequence of nodes being visited in inorder?

Any thoughts on how to implement this visit order?

Level-order Traversal

- Visit each node of Tree t as follows:
  - do something with the node / value, e.g., print it
  - visit all its children (first level in the tree) and act on the nodes
  - visit all the children's children (second level in the tree) and act on the nodes
  - visit third level in the tree, etc..

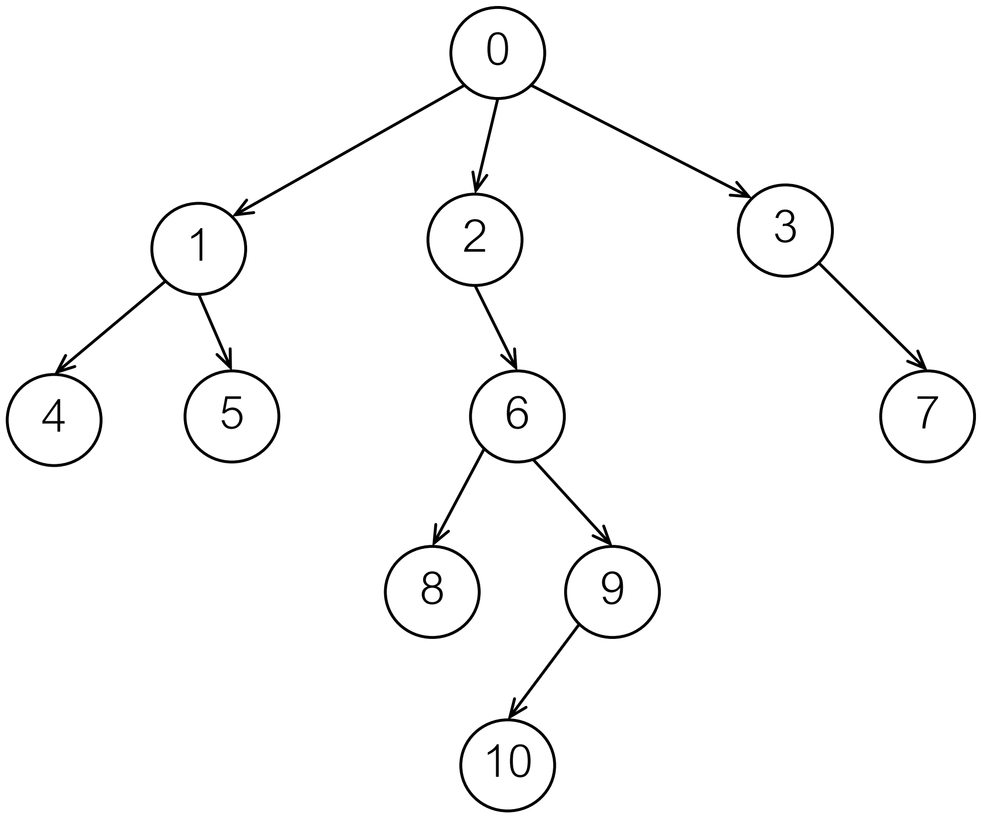
What is the sequence of nodes being visited in level order?



Level-order

- Visit each node of Tree t as follows:
    - do something with the node / value, e.g., print it
    - visit all its children (first level in the tree) and act on the nodes
    - visit all the children's children (second level in the tree) and act on the nodes
    - visit third level in the tree, etc..

What is the sequence of nodes being visited in level order?

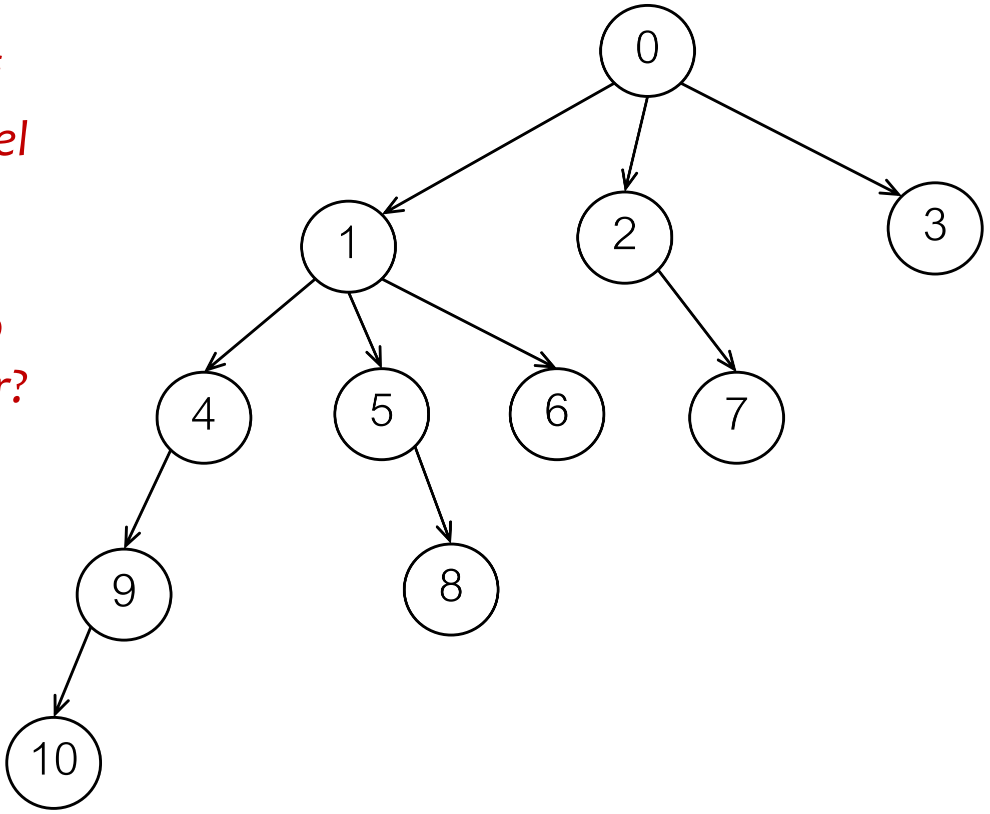


Level-order

- Visit each node of Tree t as follows:
  - do something with the node / value, e.g., print it
  - visit all its children (first level in the tree) and act on the nodes
  - visit all the children's children (second level in the tree) and act on the nodes
  - visit third level in the tree, etc..

What is the sequence of nodes being visited in level order?

Any thoughts on how to implement this visit order?



Worksheet

- Tree traversals...

Queues, Stacks, Recursion

- You may have noticed in the code for level order visit that there were no recursive calls, and a queue was used to process a recursive structure in level order
- Careful use of a stack allows you to process a tree in preorder, postorder, or inorder (no recursion needed)
- Remember recursive vs. iterative discussion from last week