CSC 236 T21: Exploring Trees

Note that this team assignment was created and updated by Dr. Jan Pearce of Berea College

Adapted POGIL Working Groups

You are to work in groups of three or four. There are typically 4 roles in this kind of activity:

a) **Facilitator** - Keeps the team on-task and on schedule.
b) **Quality Control Engineer** - Works to ensure all responses are high-quality.
c) **Process Analyst** - Considers how the team is working together and suggests ways in which it might improve.
d) **Reader/Spokesperson** - Reads the document aloud and speaks for the entire team.

See Adapted POGIL Roles for more details on these rolls.

If you have three people, combine the **facilitator and process analyst** roles. **These roles are designed to maximize communication, so we encourage you to TALK BEFORE writing** ,except when you are explicitly told you may divide and conquer.

Directions for use:

- **Note that teamworks are designed to be completed WITHOUT AI, and using AI on teamworks is prohibited because AI use will impede learning.**
- One person is to make a single copy of this worksheet for the entire group in their Google account.
- That person should share it with everyone in the group so they can all edit it.
- Please note that for all teamwork, all team members must be full participants in all of the work. For all teamwork that spans multiple days, if a member is missing on the second day of the teamwork, a copy should be made by the team so the work can be carried out separately by the smaller team using the copy, and by the missing team member using the original document.
- The reader/spokesperson changes the document title so that it is *csc236_t21_yourusernames*

Pick a role **YOU HAVE NOT HAD A CHANCE TO DO RECENTLY**:

Team Roles	Member Name
Facilitator	
Quality Control Engineer	
Process Analyst	
Reader/Spokesperson	
TEAM NAME	

Trees

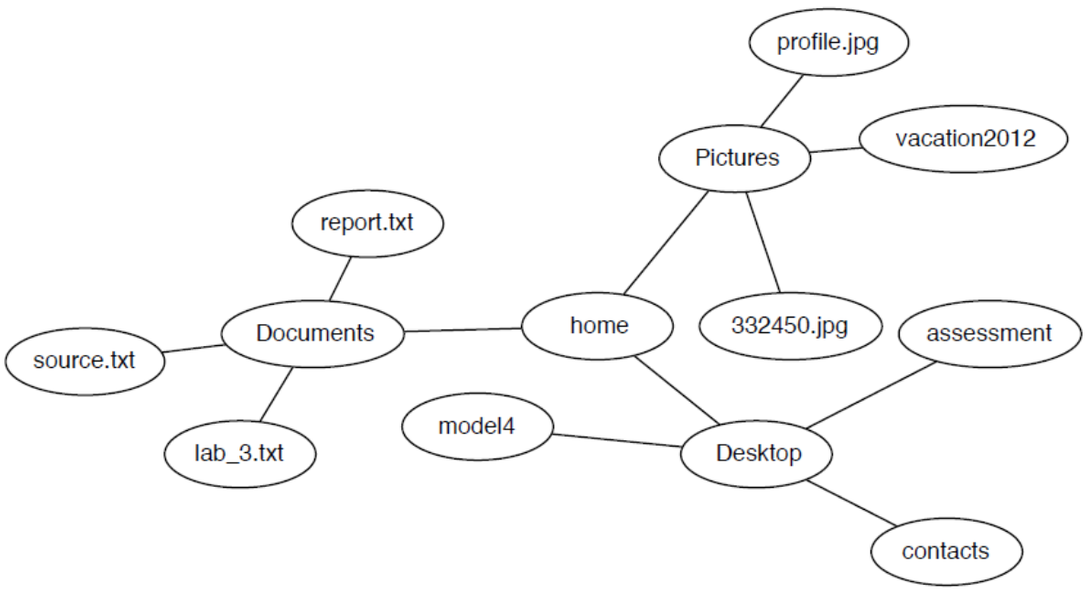
A **tree** is a widely used data structure that simulates a hierarchical tree structure, with a root value and a set of linked nodes which are each subtrees. Note that an important property of a

tree structure is that if you start at the root node, there is a unique path to any given node in the tree.

A **binary tree** is a tree in which each node has a value and at most two child nodes, usually called the left child and right child, which are themselves the roots of subtrees.

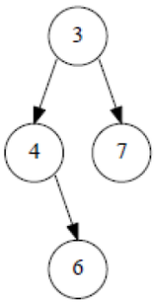
A **binary search tree** (**BST**) is just a binary tree with the extra property that holds for every node in the tree: the values in the left subtree are less than the value at that node, and the values in the right subtree are greater than the value at that node.

Consider the following image:



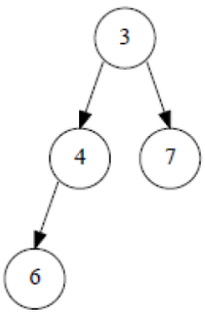
1. The above is not drawn as we expect a tree, but could it be represented as a tree if directions were added to all of the edges? Why or why not?	
2. Ignoring the meanings of the node names, if it can be represented as a tree, which of the nodes ***could*** *be* used as the root node? Identify all of the nodes that ***could*** *be* the root.	
3. Looking at the node names, does one node make the most sense as the root?	
4. If directions were added to all of the edges, could the above be a binary tree? Why or why not?	
5. If directions were added to all of the edges, could the above be a binary search tree? Why or why not?	

Consider:



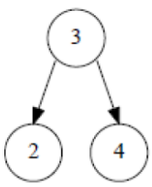
6. Is the above represented as a tree? a binary tree? a binary search tree? Explain why or why not for each. (Be sure to carefully read the definitions given above.)	

Consider:



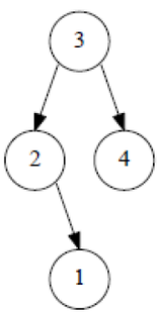
7. Is the above represented as a tree? a binary tree? a binary search tree? Explain why or why not for each.	

Consider:



8. Is the above represented as a tree? a binary tree? a binary search tree? Explain why or why not for each.	

Consider:



9. Is the above represented as a tree? a binary tree? a binary search tree? Explain why or why not for each.	

Tree Traversals

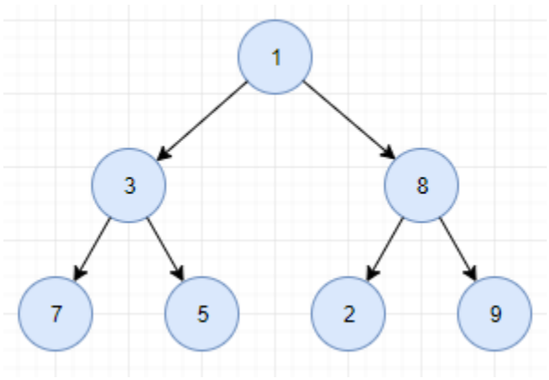
The **depth** of a node indicates how many edges are between it and the root node. The root node has a depth of zero. The **height** or depth of a tree is the maximum depth of all nodes in the tree.

A **tree traversal** is a specific order in which to trace the nodes of a tree. There are 3 common tree traversals.

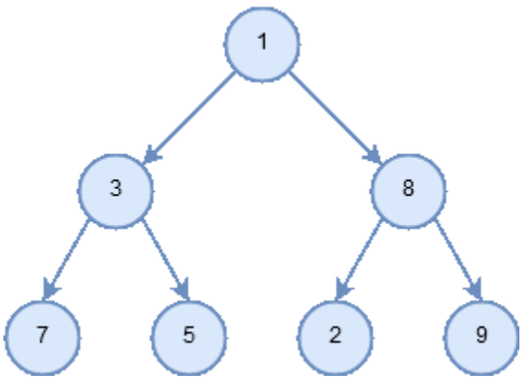
1. in-order: left, root, right
2. pre-order: root, left, right
3. post-order: left, right, root

This order is applied recursively.

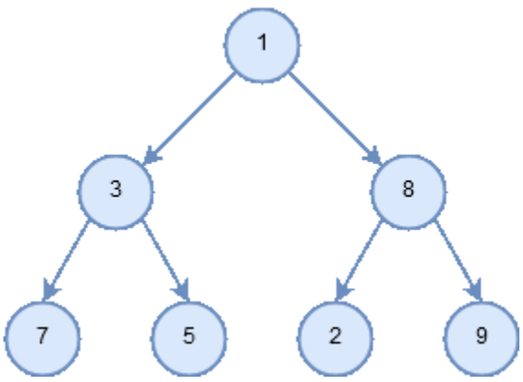
Consider the following example:



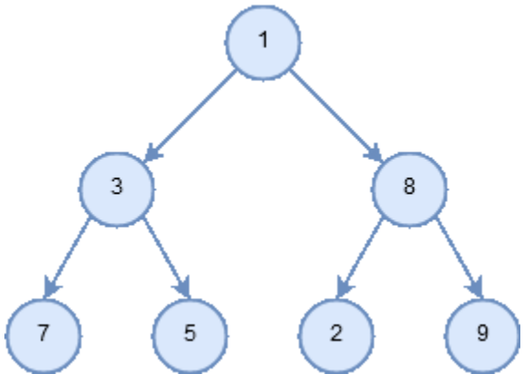
1. Depth of node 3: 1
2. Height of tree: 2



In-order traversal (left, root, right): 7, 3, 5, 1, 2, 8, 9

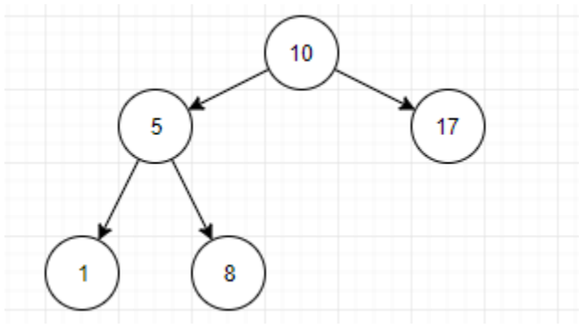


Pre-order traversal (root, left, right): 1, 3, 7, 5, 8, 2, 9



Post-order traversal (left, right, root): 7, 5, 3, 2, 9, 8, 1

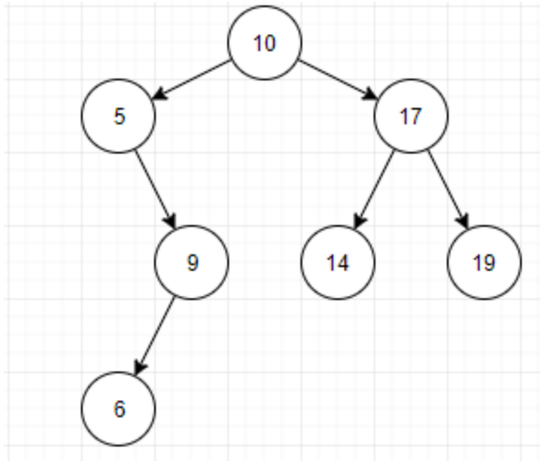
One more example of traversals without the animation



This tree gives the following:

Preorder (Root, Left, Right) will give 10 5 1 8 17
Inorder (Left, Root, Right) will give 1 5 8 10 17
Postorder (Left, Right, Root) will give 1 8 5 17 10

10. Now, consider the binary search tree given below:

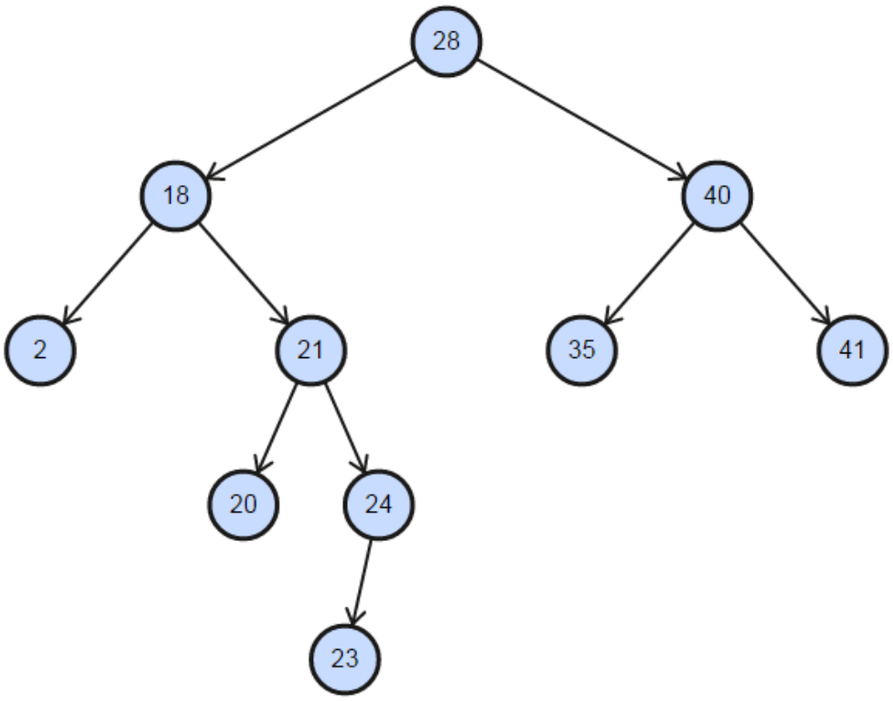


.

Fill in the traversals

Preorder (Root, Left, Right):
Inorder (Left, Root, Right):
Postorder (Left, Right, Root):

Consider the bigger tree:



11. What is the height of the above tree?	
12. List the order of nodes visited in an in-order traversal of the above.	
13. List the order of nodes visited in an pre-order traversal of the above.	
14. List the order of nodes visited in an post-order traversal of the above.	

Real-World Applications

Consider the following situations and determine which data structure seems most appropriate:

Application	**True/False Can be easily represented as a tree data structure, and a brief explanation.**
An operating system maintains a disk's file system as a tree where file folders act as tree nodes.	**True:** because all folders are subfolders of the “Computer” and all folders are wholly inside of another folder so their contents are subtrees.
15. The biological taxonomy is a tree in which each level can be named: Kingdom, Phylum, Class, Order, Family, Genus, Species.	
16. Kentucky can be represented by a tree in which cities are nodes and roads are edges.	
17. The Internet can be represented as a tree structure in which each page is a node and the links are the edges.	
18. The chapters, sections, and subsections of a book.	

Summary Learning

19. Discuss the most significant things you have learned about the tree data structure.

Submission and Integrity

- Before submission, please ensure that all names at the top fully participated in the entire teamwork. Please note that if a teammate was absent for part of the teamwork, they must not be given credit for work they did not complete. If a team member was not present for part of the work, this must be explained at the top, and your team will have completed the work in a copy. If you forgot, you can make a copy now, and then undo all of the changes in the original back to the appropriate point for the missing teammate to complete.
- All team members must download a copy of this document as a PDF and upload it to Moodle.
- Note that by uploading this document to Moodle, you are asserting that this is fully the work of the team members listed at the top, and that you have not used any AI tools except those built into the IDE.