

CSCA48 Winter 2018

Week 5: Trees

Marzieh Ahmadzadeh, Nick Cheng
University of Toronto Scarborough

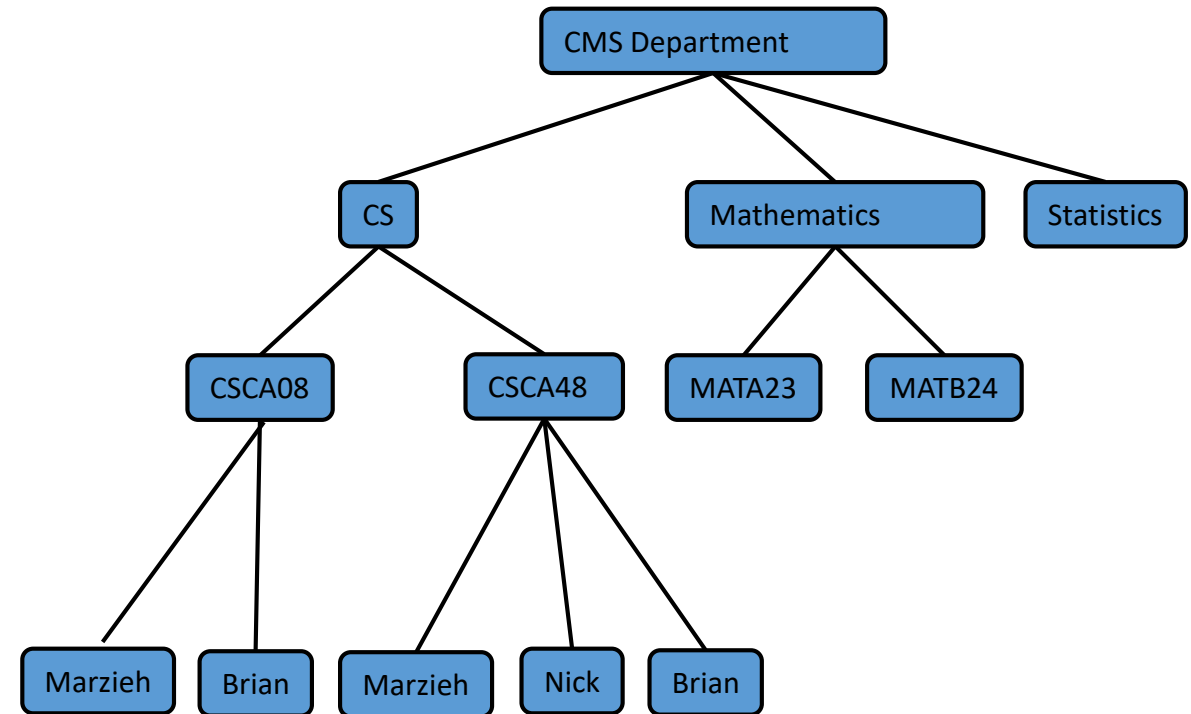


Administrative Detail

- First assignment
 - Due date
 - Do it before the exam

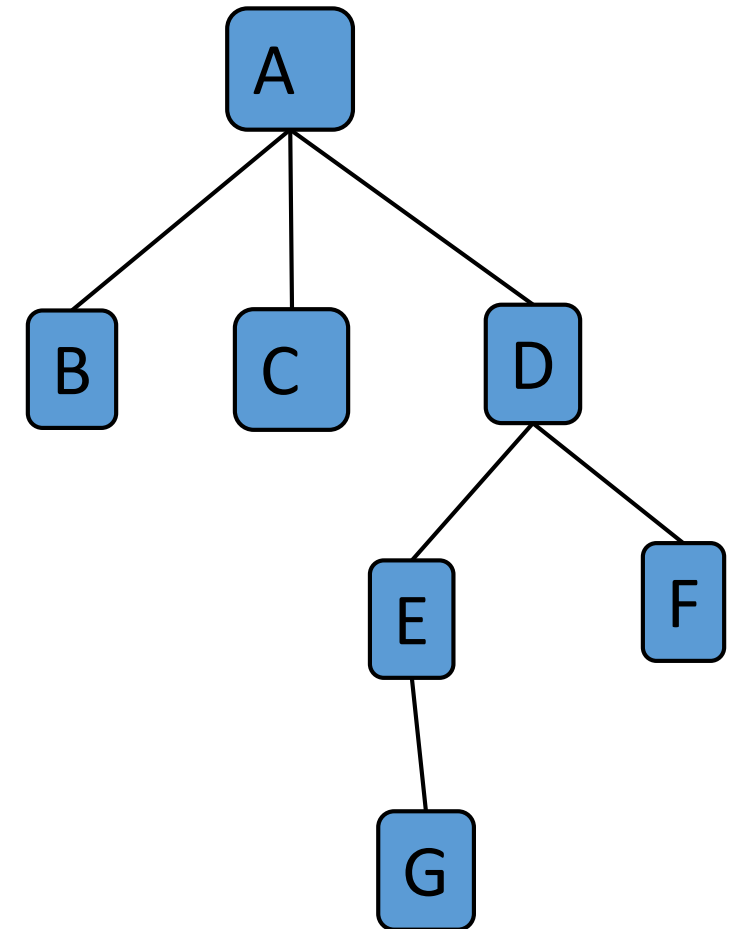
Non-Linear Data Structure

- Not all data can be represented by a linear data structure
- Application:
 - Organization charts
 - Family Tree
 - File systems
 - Programming environments
 - Decision Making
 - Games



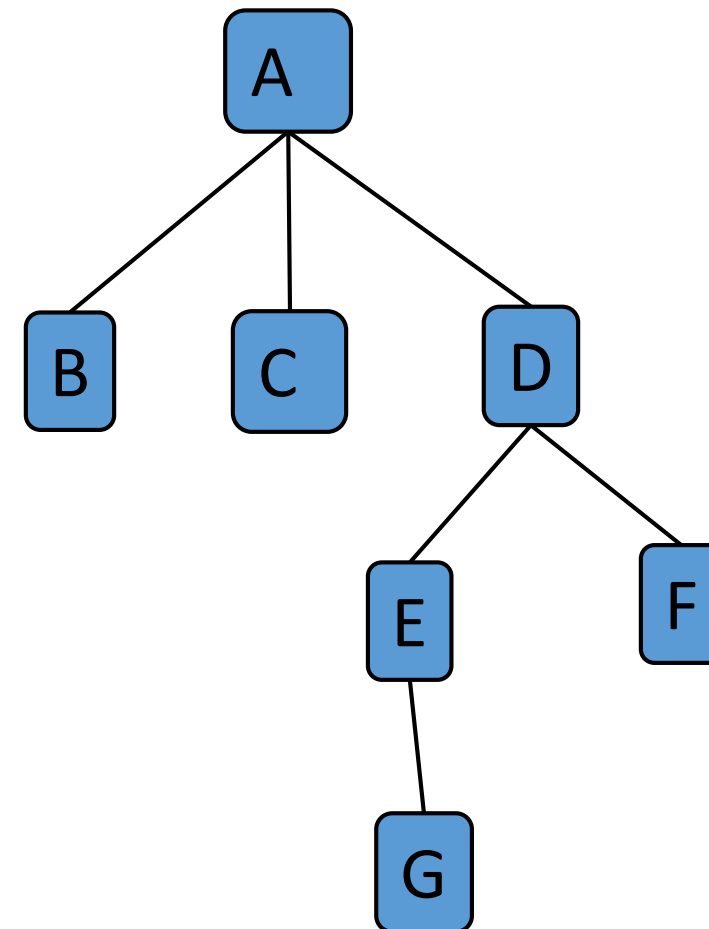
Trees

- A tree T
 - is a non-linear data structure
 - consists of a set of nodes storing elements with a parent-child relationship
- Every node has a unique parent.
 - except one that has no parent.
- Each node may/may not have children



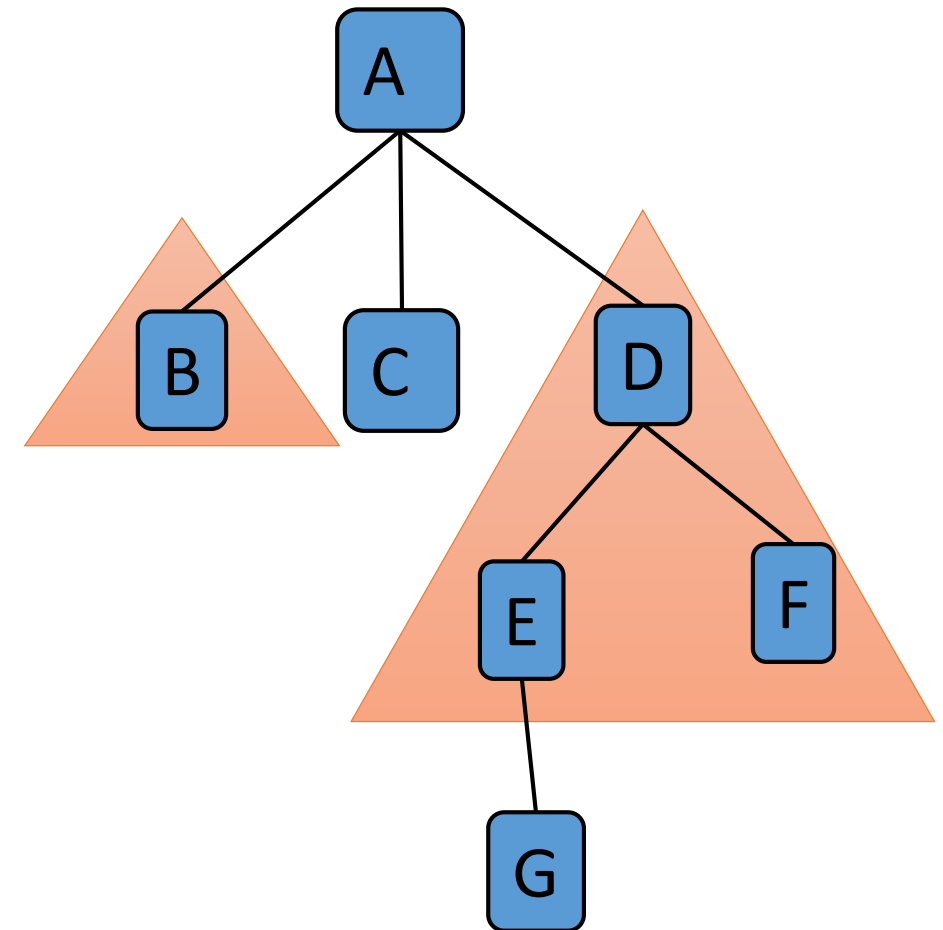
Terminology

- Root: node without any parent (A)
- Internal node: a node with at least one child (A, D, E)
- External node (AKA leaf): a node without children (B, C, G, F)
- Depth of a node: number of ancestors
 - Remember that each node has only one parent.
- Height of a tree: $\text{Max}(\text{depth of all nodes})$



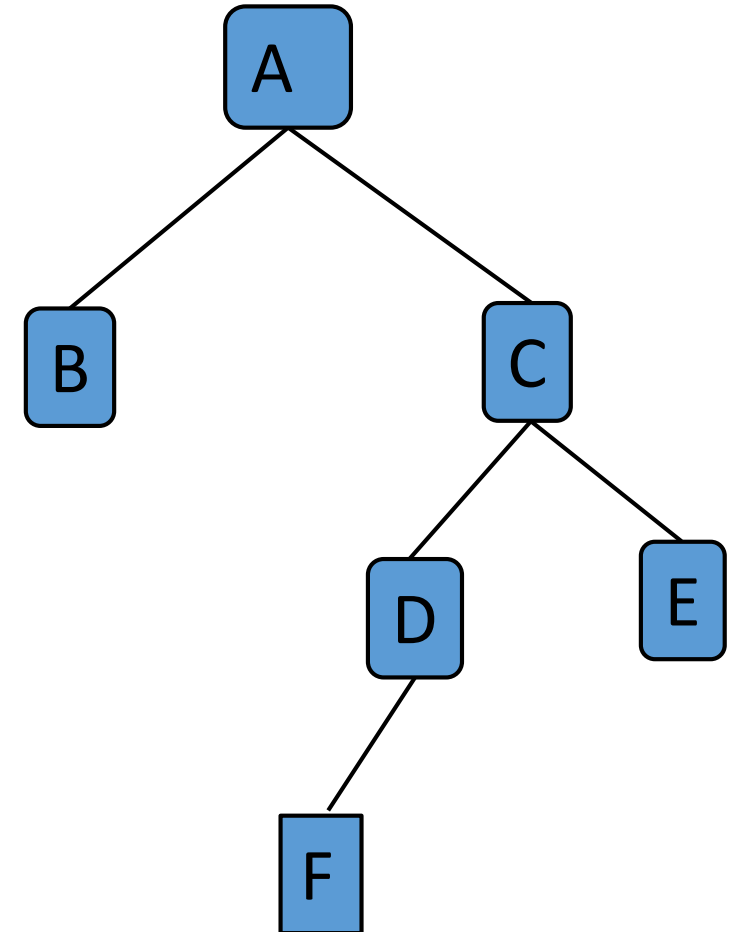
Terminology

- Subtree: a tree consisting of a node and its descendants
- Siblings: nodes that share the same parent ([B,C,D], [E,F])
- Edge: a pair of nodes such that they have parent-child relationship.
- Path: a sequence of consecutive nodes



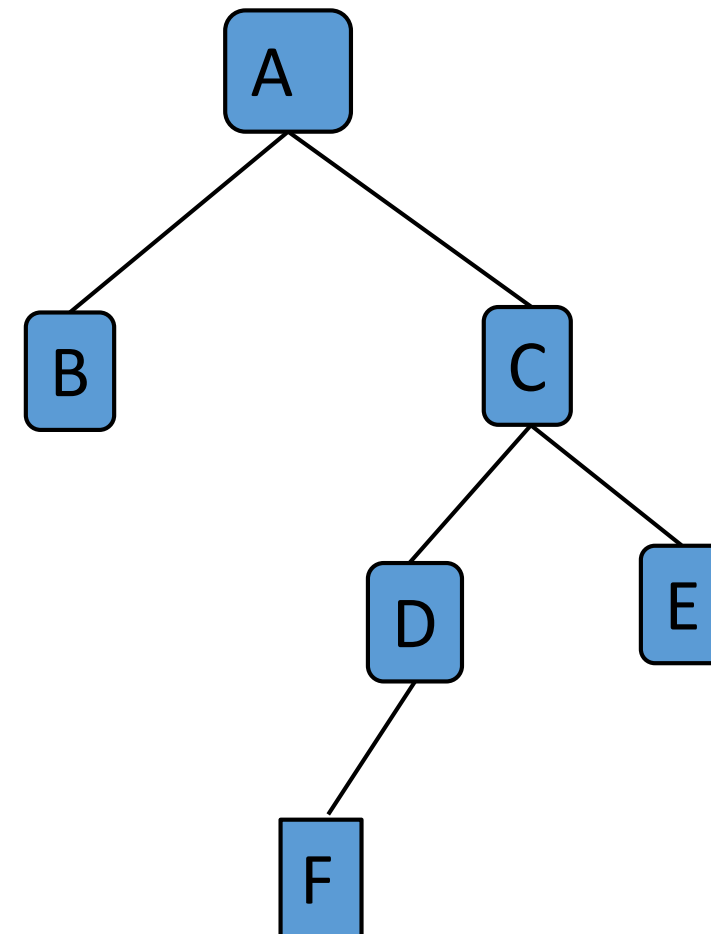
Binary Tree

- A BT is a tree in which
 - Each node has at most two children
 - Each child node is labeled as a `left` or `right` child



Binary Tree Traversal

- Breadth First
 - Start from the root: visit all the nodes at depth i before visiting the nodes at depth $i + 1$. (A B C D E F)
- Depth First
 - Start from the root: visit all the nodes in a path until you get to a leaf. (A B C D F E)
- Pre-order
 - Start from the root: visit the root, visit the left subtree, visit the right subtree. (A B C D F E)
- In-order
 - Start from the root: visit the left subtree, visit the root, visit the right subtree. (B A F D C E)
- Post-order
 - Start from the root: visit the left subtree, visit the right subtree, visit the root. (B F D E C A)



Example

- Draw a binary tree that satisfies the followings simultaneously:
 - Pre-order traversal: ABCDEFG
 - In-order traversal: DCEBFAG
- Post-order traversal: TREESAREFUN
- In-order traversal: TRSEENARUEF

Binary Search Tree

- Is a binary tree that satisfies the following property:
 - Let u , v , and w be three nodes such that
 - u is the left child of v
 - w is the right child of v .
 - Then
$$data(u) \leq data(v) \leq data(w)$$
- In-order traversal of a BST, returns a sorted list