

Some more data structures

Lecture 12

Sets

- What is a set?
 - What is a multiset?
 - Refer to STL set in C++.
 - It requires the `<set>` header file.
-
- <https://www.cplusplus.com/reference/set/set/>

Operations on Sets

- Some operations for sets:
 - subset
 - union
 - intersection
 - difference
 - element

Maps

- Maps have a key value pairing
- Refer to map and multimap in the STL C++
- <https://www.cplusplus.com/reference/map/map/>

Deque

- A Deque (double-ended queue) allows insertion and deletion at both ends
- Deque or Double Ended queues are sequence containers with the feature of expansion and contraction on both ends.
- <https://www.geeksforgeeks.org/deque-front-deque-back-cpp-stl/>
- <https://www.cplusplus.com/reference/deque/deque/>

Binary Trees

- Definition: a binary tree T is either empty or has these properties:
 - Has a root node
 - Has two sets of nodes: left subtree L_T and right subtree R_T
 - L_T and R_T are binary trees

Binary Trees (cont'd.)

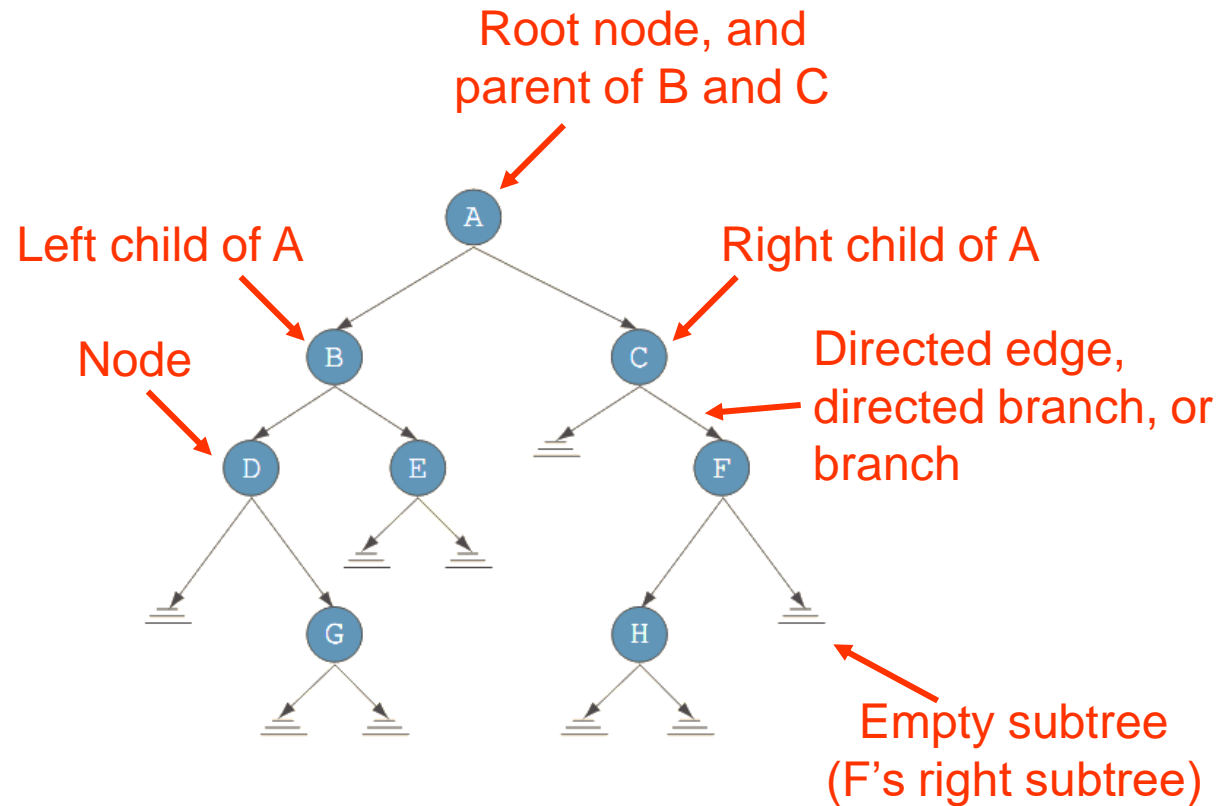


FIGURE 19-1 Binary tree

Binary Trees (cont'd.)

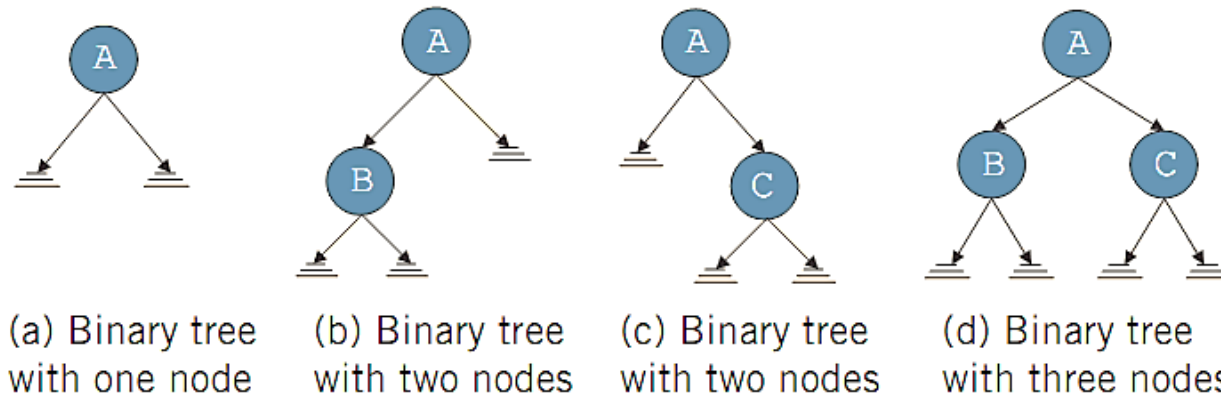


FIGURE 19-2 Binary tree with one, two, or three nodes

Binary Trees (cont'd.)

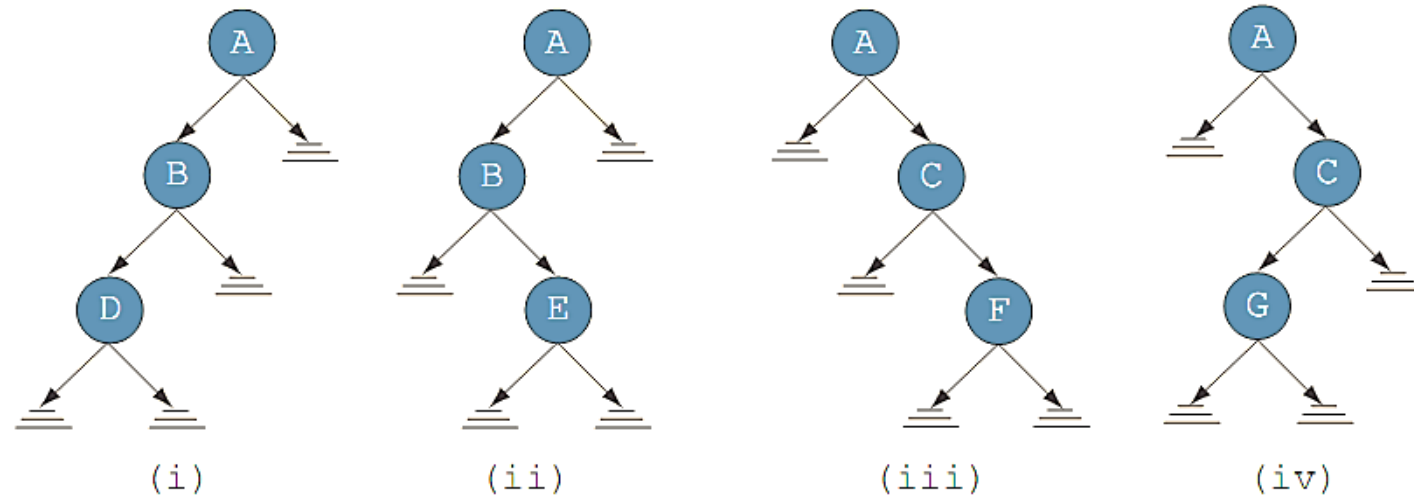


FIGURE 19-3 Various binary trees with three nodes

Binary Trees (cont'd.)

- Every node has at most two children
- A node:
 - Stores its own information
 - Keeps track of its left subtree and right subtree using pointers
 - `lLink` and `rLink` pointers

Binary Trees (cont'd.)

- A pointer to the root node of the binary tree is stored outside the tree in a pointer variable

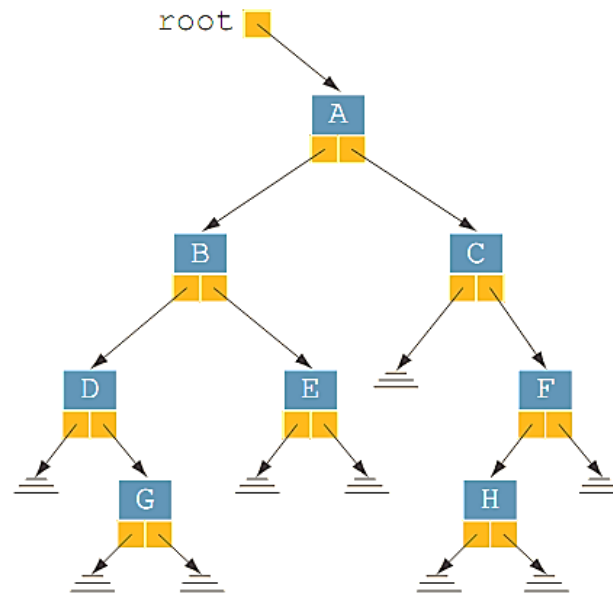


FIGURE 19-4 Binary tree

Binary Trees (cont'd.)

- Leaf: node that has no left and right children
- U is parent of V if there is a branch from U to V
- There is a unique path from root to every node
- Length of a path: number of branches on path
- Level of a node: number of branches on the path from the root to the node
 - Root node level is 0
- Height of a binary tree: number of nodes on the longest path from the root to a leaf

Binary Tree Traversal

- Insertion, deletion, and lookup operations require traversal of the tree
 - Must start at the root node
- Two choices for each node:
 - Visit the node first
 - Visit the node's subtrees first

Binary Tree Traversal (cont'd.)

- Inorder traversal
 - Traverse the left subtree
 - Visit the node
 - Traverse the right subtree
- Preorder traversal
 - Visit the node
 - Traverse the left subtree
 - Traverse the right subtree

Binary Tree Traversal (cont'd.)

- Postorder traversal
 - Traverse the left subtree
 - Traverse the right subtree
 - Visit the node
- Listing of nodes produced by traversal type is called:
 - Inorder sequence
 - Preorder sequence
 - Postorder sequence

Binary Search Trees

- Binary trees are trees where
 - every node has 1 piece of data and two pointers
 - therefore every node has 0:2 children
- Binary search trees are binary trees where
 - every node has data that is greater than the data in all nodes to the left of it.
 - every node has data that is less than the data in all nodes to the right of it.
- Note that this contrasts with the heap, where a node's data was always guaranteed to be less than (for a min-heap) or greater than (for a max-heap) all data in its subtree. [1]
- Since the data sorting is based on a unique key, there are never two identical sets of data.