ALGONQUIN COLLEGE

CST8130 – Data Structures

Professor : Dr. Anu Thomas
Email: thomasa@algonquincollege.com
Office: T314

Trees

Testing your program for every case

What were the standard tests for a linked list in a test plan?

What would the standard tests be for a (binary) tree in a test plan?

Linked list tests

Check “change” works for first element (head and/or tail must be updated)

Check it works in middle– prev and/or next links on both sides must be updated

check it works at end (tail must be updated)

Check it works if list is empty

Check is works if list is full (if applicable)

Check it works if element is not in list

Tree tests

Check “change” works if its to the first node (root must be updated)

Check change works if it’s a node to left (with leaves)

Check change works if it’s a node to the right (with leaves)

Check change works if it affects a leaf

Check change works to empty tree

Check change works if not in tree (if applicable)

How to guarantee O(log n) search in binary tree

Tree must be complete

- for any node, height of left subtree and right subtree does not differ by more than 1.

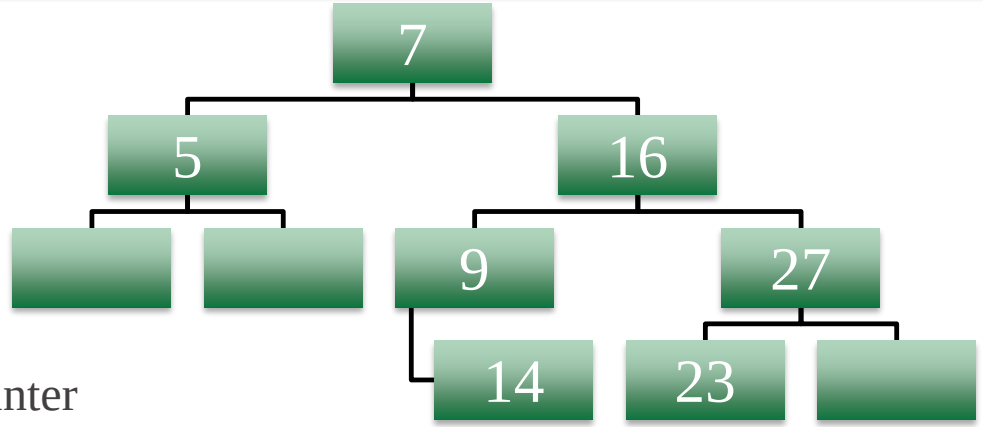
Delete a Node

Four possibilities:

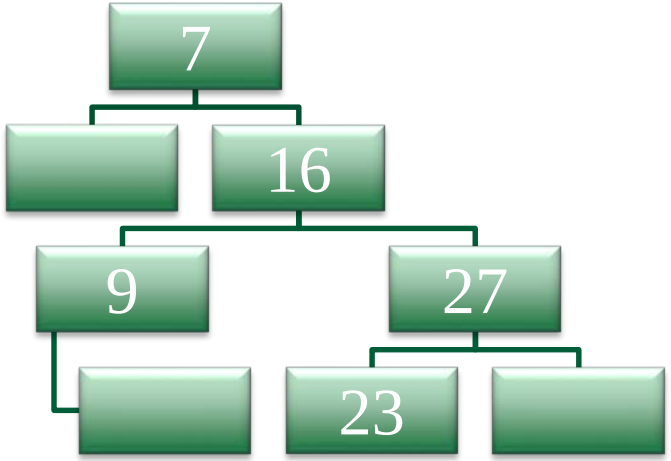
1. Node to be deleted has no children
2. Node to be deleted has subtree on left but no subtree on right
3. Node to be deleted has subtree on right but no subtree on left
4. Node to be deleted has subtree on right and subtree on left

Node to be deleted has no children

- Set parent's pointer (left or right) to NULL
- Delete object



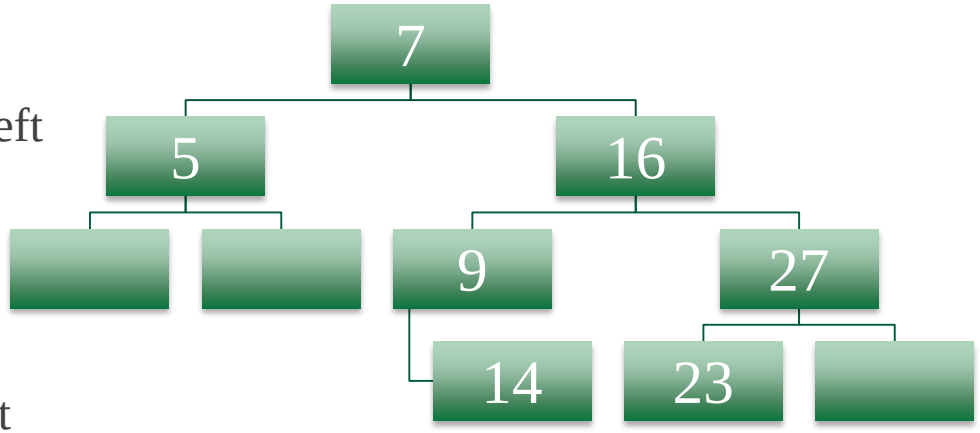
- Example – delete 14, set 9's right pointer to NULL
- Example – delete 5, set 7's left pointer to NULL

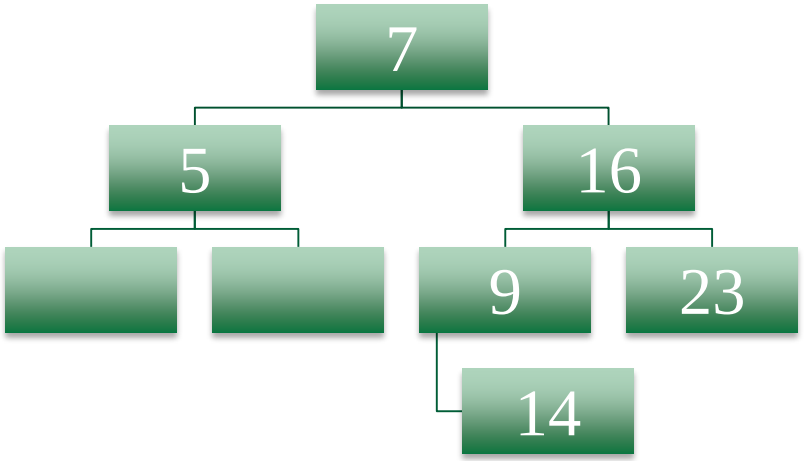


Node to be deleted has subtree on left but no subtree on right

- Attach left subtree to parent's pointer (left or right)
- Delete object

Example – delete 27, attach 23 to 16's right pointer



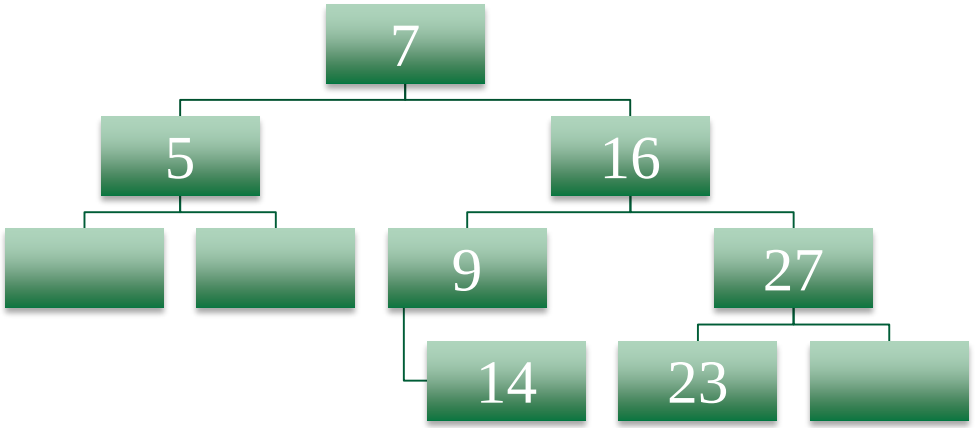


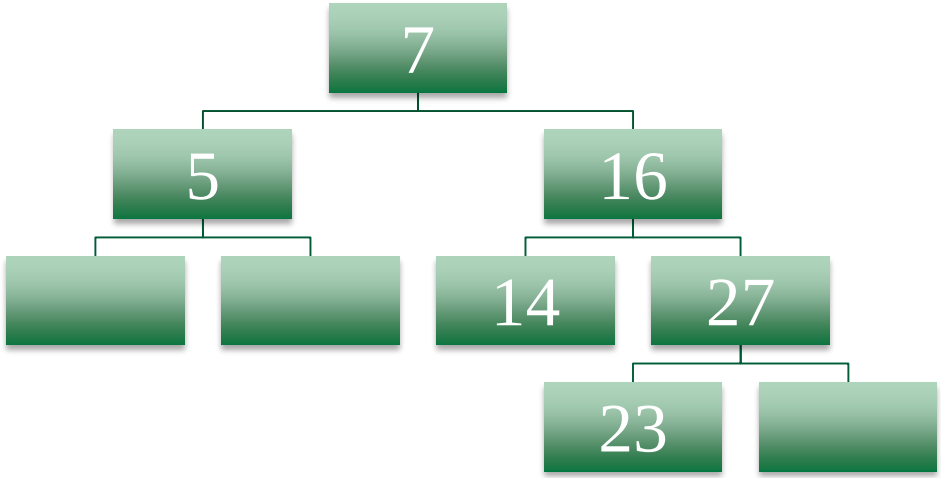
Node to be deleted has subtree on right but no subtree on left

Attach right subtree to parent's pointer (left or right)

Delete object

Example – delete 9, attach 14 to 16's left pointer

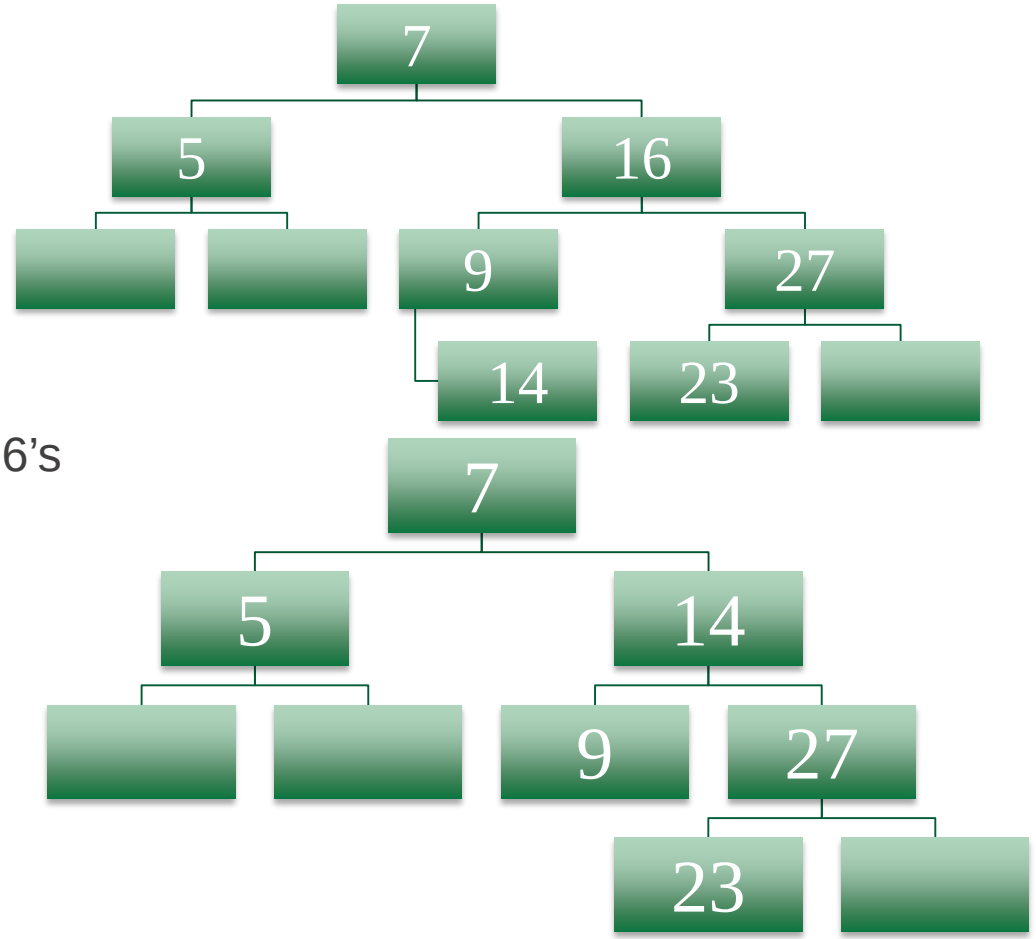




Node to be deleted has subtree on right and subtree on left

- Find right-most node on left subtree
- Move data to replace deleted nodes data
- Delete right-most object

Example – delete 16, move 14 to 16's data, delete 14



Questions?