CMPUT 274

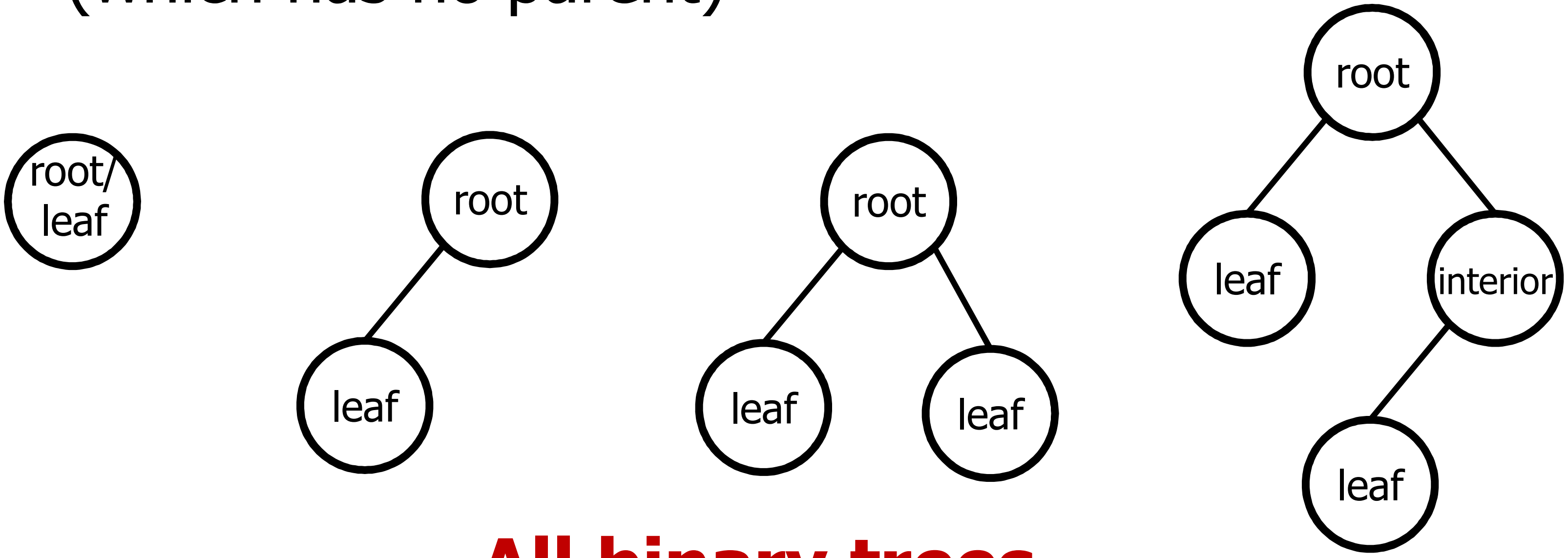
Binary Tree Implementation

Topics Covered:

- Recursive representation
- Tree Leaf class
- Tree branch class
- Binary tree traversal

Defining a Binary Tree

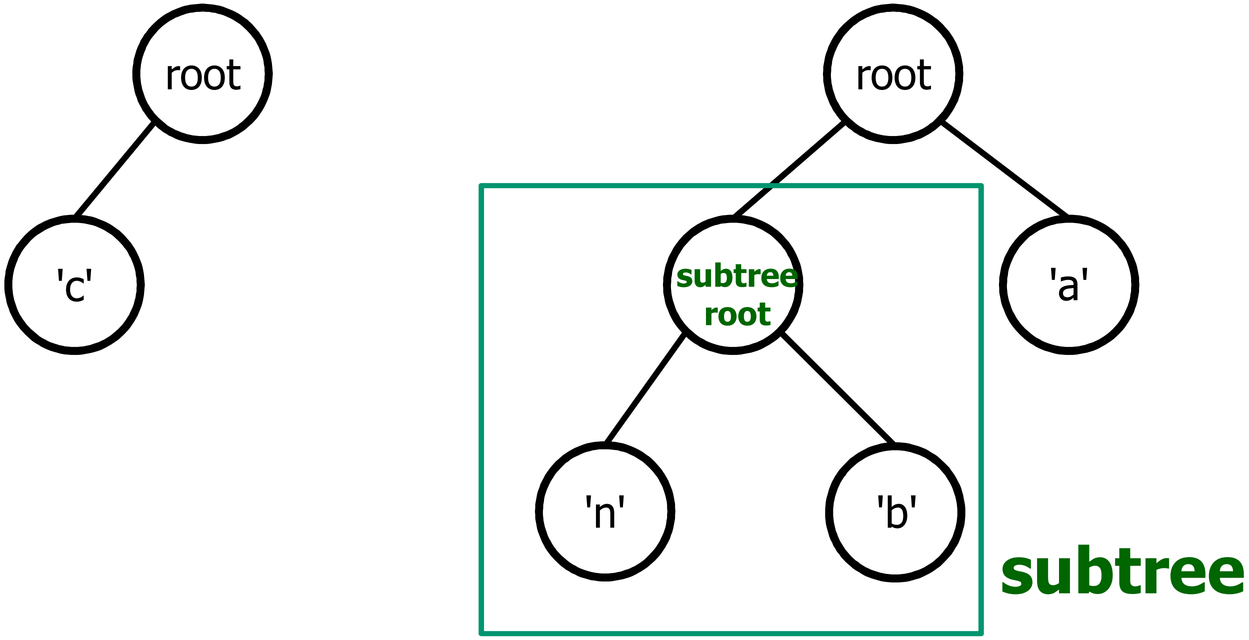
- Recall:
  - Binary tree is made up of 1 or more nodes
  - Each node has 0, 1, or 2 children
  - All nodes have 1 parent, except the root node (which has no parent)



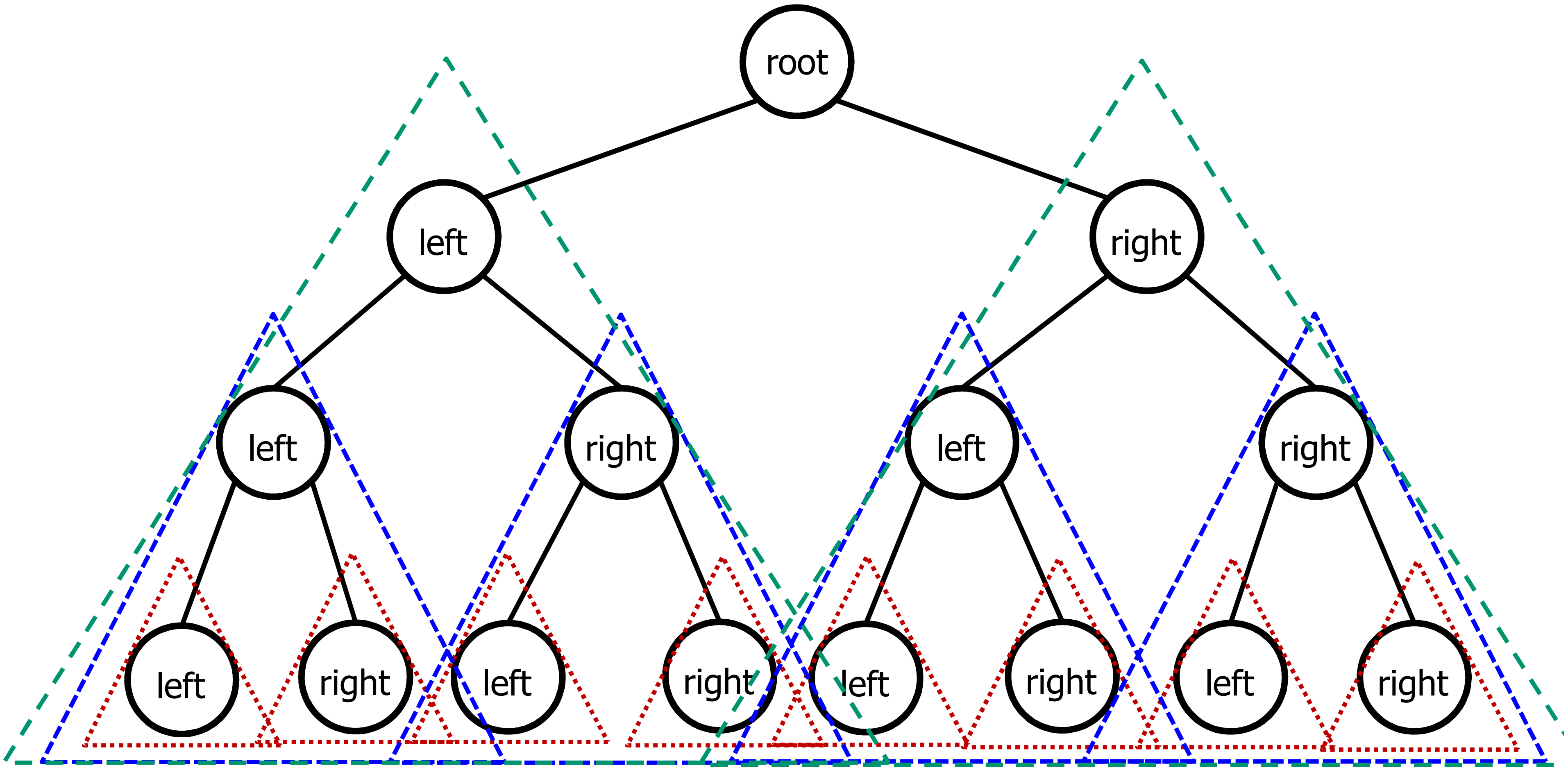
All binary trees

Defining a Huffman Tree

- In Huffman tree:
  - All leaf nodes store a value: byte to be compressed
  - All interior nodes are the root node of subtree



Recursive Definition



How to Implement in Python?

- Need a way to represent the tree nodes, and the relationships between them

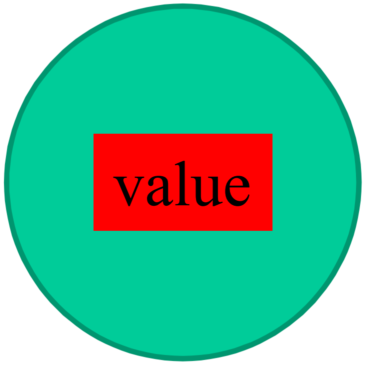
- Option 1: List of lists
  - Each subtree is a list that contains the root, the left subtree, and the right subtree

    → gets complicated quickly; hard to keep track of all of the nested subtrees

- Option 2: Custom classes
  - Tree branch class capable of containing left subtree and right subtree
  - Tree leaf class to represent the individual leaf nodes

Tree Leaf Class

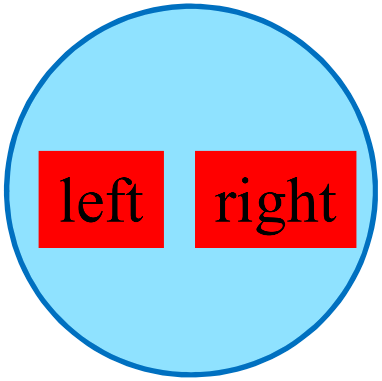
- Tree leaf properties:
  - Has a value (uncompressed byte) that it is storing



- Tree leaf behaviours:
  - N/A

Tree Branch (Subtree) Class

- Tree branch properties:
  - Left child
  - Right child



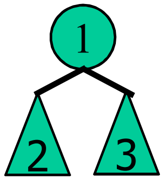
- Tree branch behaviours:
  - N/A

Binary Tree Traversals

- There are four common binary tree traversals:

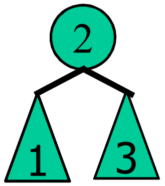
Preorder: process root then left subtree then right subtree

Root (Left) (Right)



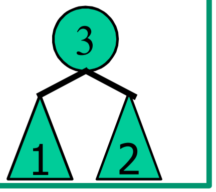
Inorder: process left subtree then root then right subtree

(Left) **Root** (Right)



Postorder: process left subtree then right subtree then root

(Left) (Right) **Root**

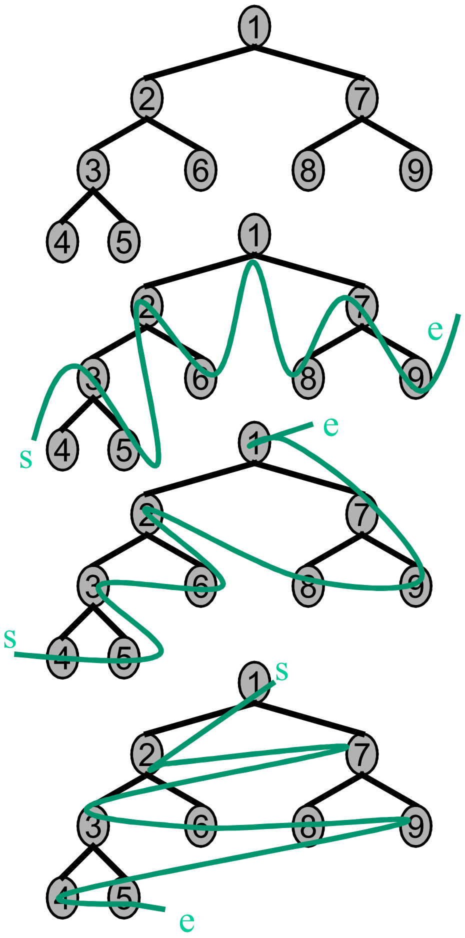


Levelorder: process nodes of level i, before processing nodes of level i + 1, etc

- Processing of left and right subtrees is done recursively

Binary Tree Traversals: Example

- Preorder: 1 2 3 4 5 6 7 8 9
- Inorder: 4 3 5 2 6 1 8 7 9
- Postorder: 4 5 3 6 2 8 9 7 1
- Levelorder: 1 2 7 3 6 8 9 4 5


1
2
7
3
6
8
9
4
5
s
e
e
s
s
e