PROGRAMMING AND DATA STRUCTURES

BINARY TREES

HOURIA OUDGHIRI

SPRING 2023

OUTLINE

- Binary trees
- Properties of binary trees
- Traversal of binary trees

STUDENT LEARNING OUTCOMES

At the end of this chapter, you should be able to:

- Describe the properties of binary trees
- Traverse binary trees using preorder, ignorer, and postorder traversals

What is a binary tree?

- Data organized in a binary tree structure
- Easy and efficient access and update in large collections of data
- Used for efficient search operations
- Wide range of applications: mathematical expressions, game strategies, decision trees, data compression, …

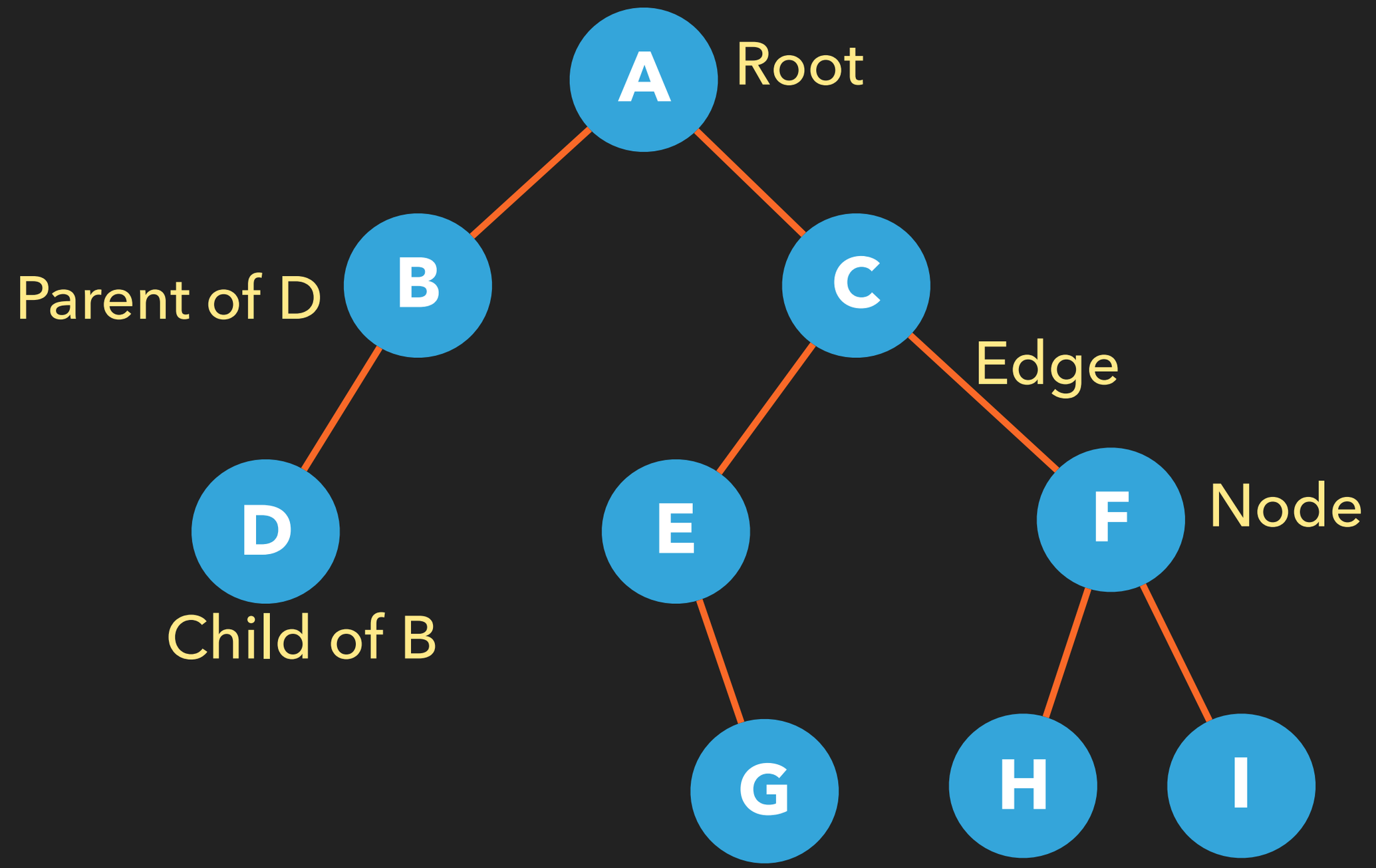
Properties

- Set of elements called nodes (vertices) interconnected with edges (arcs)
- The first node is called the root
- The root is connected to two binary trees (left subtree and right subtree)

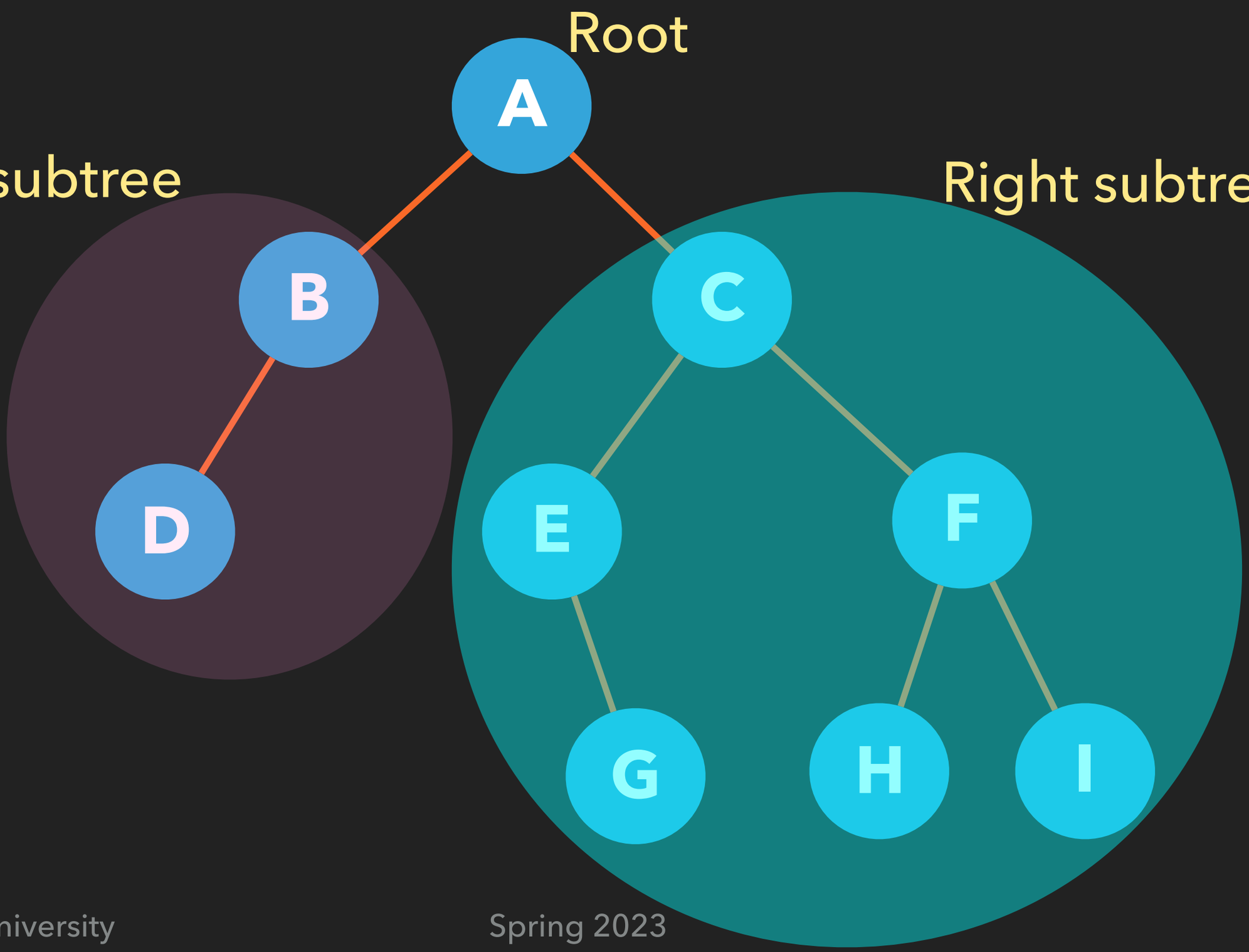
Properties

- Every node has a parent (except the root) and may have no child, one child or two children at most
- The root is the ancestor of all the nodes in the tree

Properties



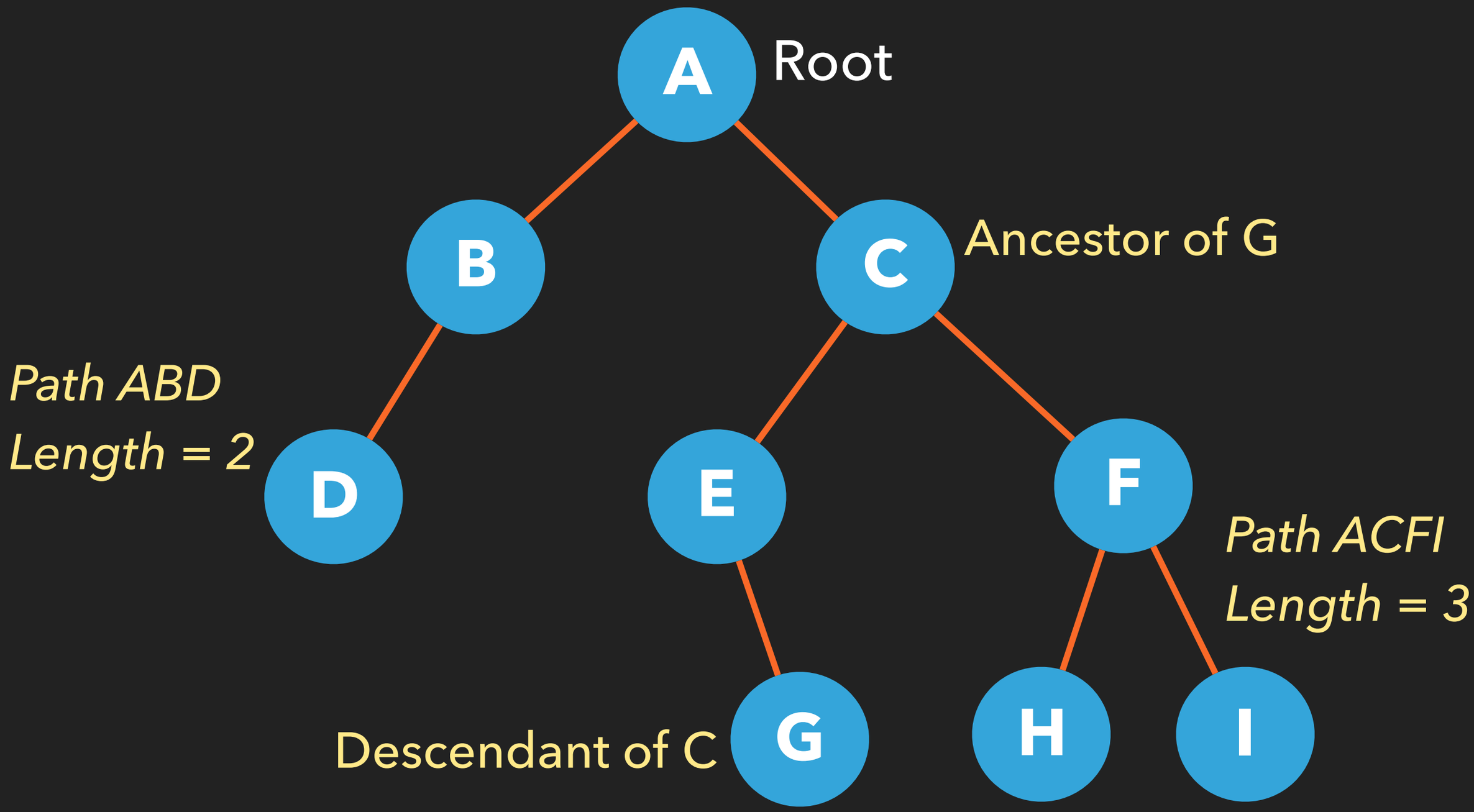
Properties



Properties

- Path: Sequence of connected nodes starting at any level of the tree
- Length of a path: the number of nodes in the sequence - 1 (number of edges)
- If there is a path from node P to node Q, Q is the descendant of P and P is an ancestor of Q

Properties

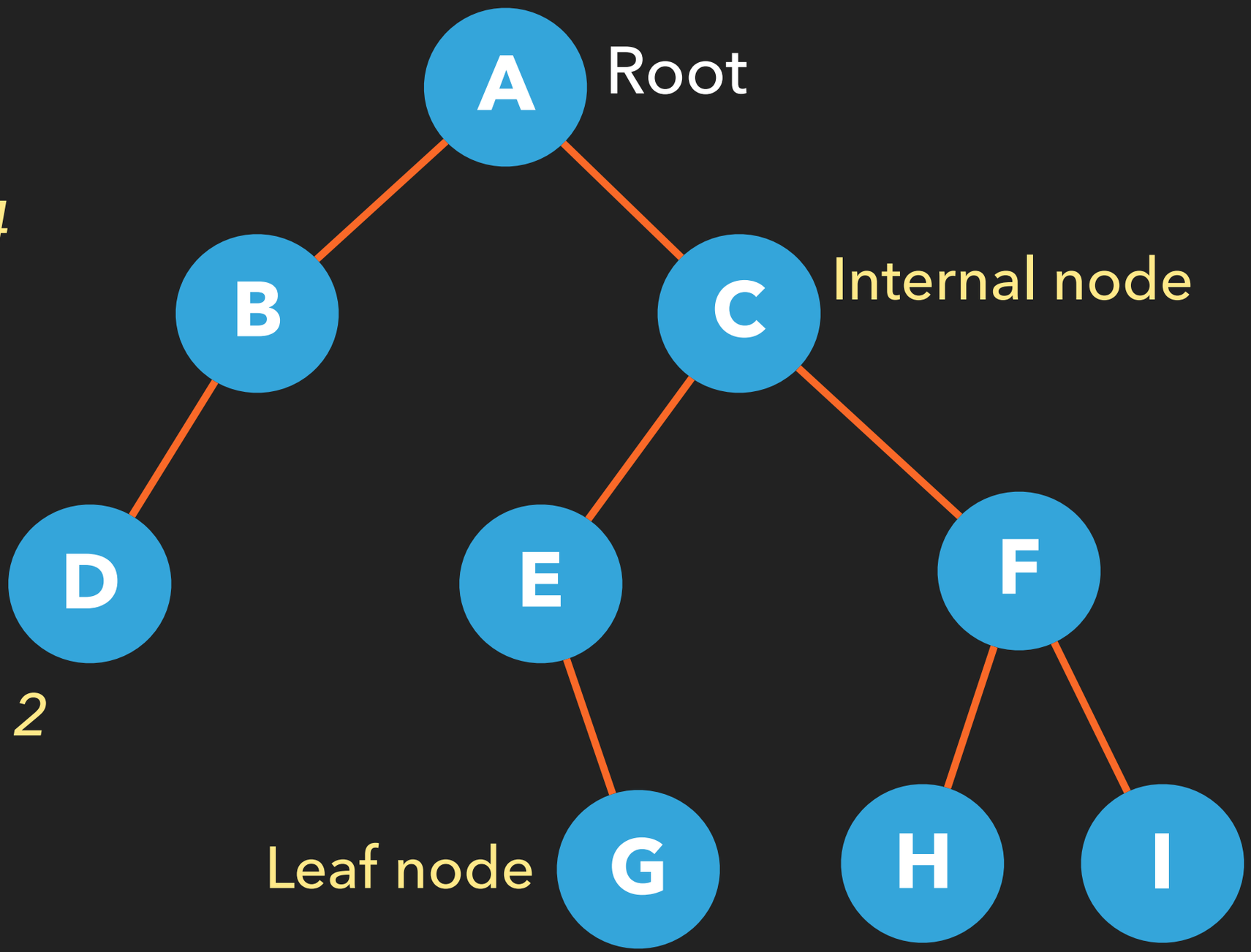

A
Root
B
C
Ancestor of G
Path ABD
Length = 2
D
E
F
Path ACFI
Length = 3
Descendant of C
G
H
I

Properties

- Depth of a node: Length of the path from the root to the node
- Height of a tree: the depth of the deepest node + 1
- Leaf node: node with no children
- Internal node: node with at least one child

Properties

Height = 4



Depth(D) = 2

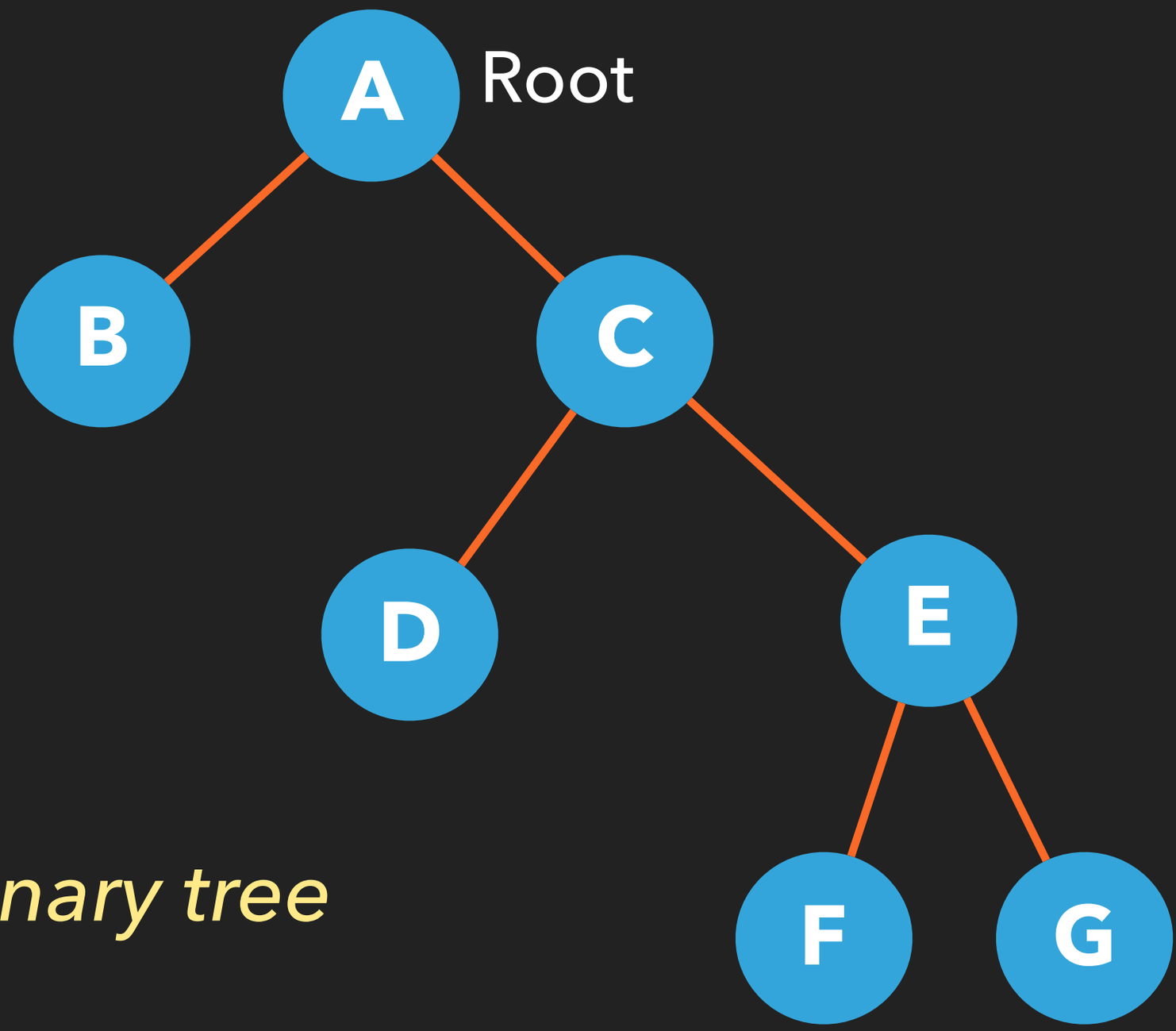
Depth(H) = 3

Properties

- **Full Binary Tree**: each node is a leaf or an internal node with exactly two children

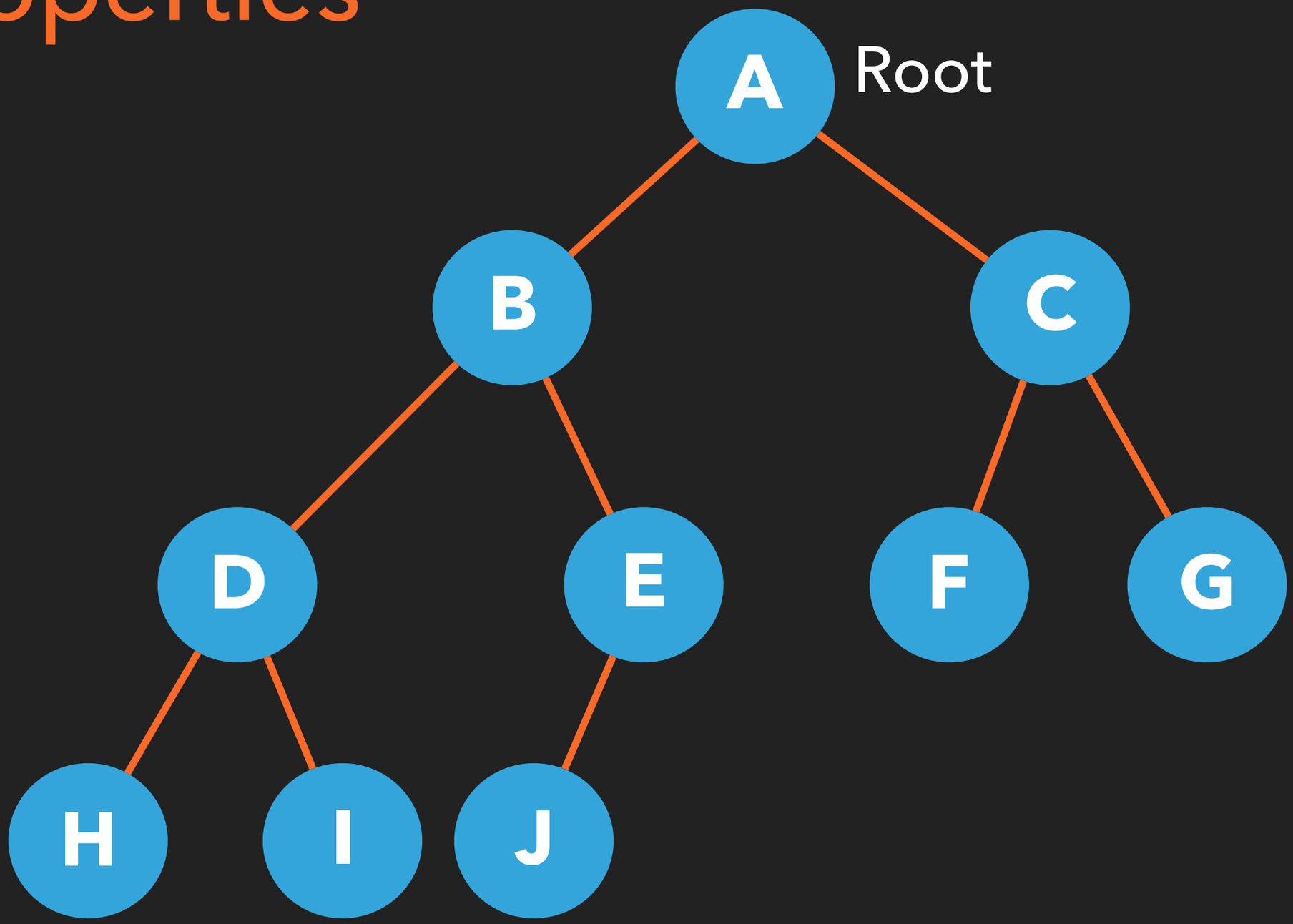
- **Complete Binary Tree**: Every level is filled (two children) except the last level, and the leaves on the last level are placed leftmost

Properties



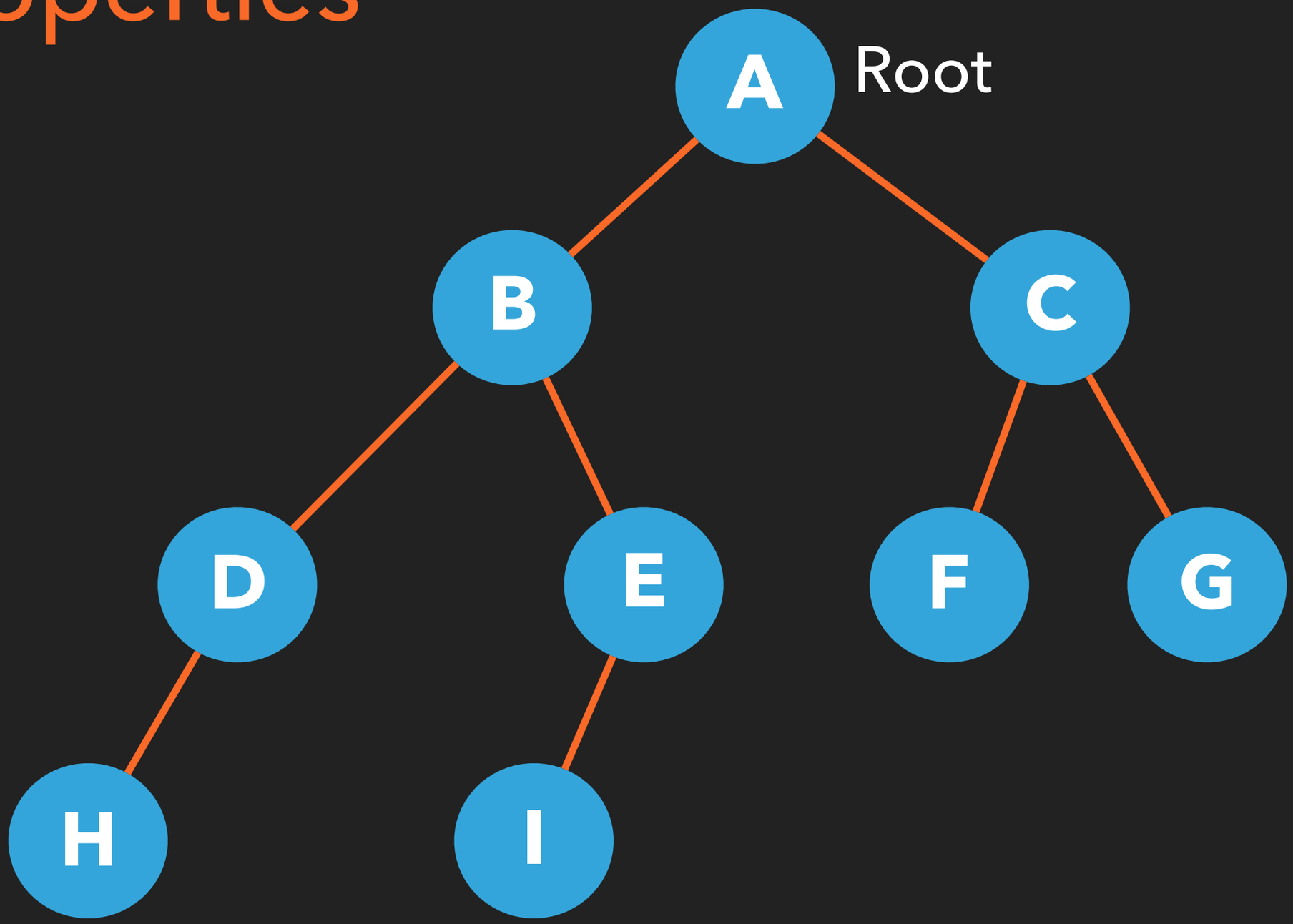
Full binary tree

Properties



Complete binary tree but not full

Properties

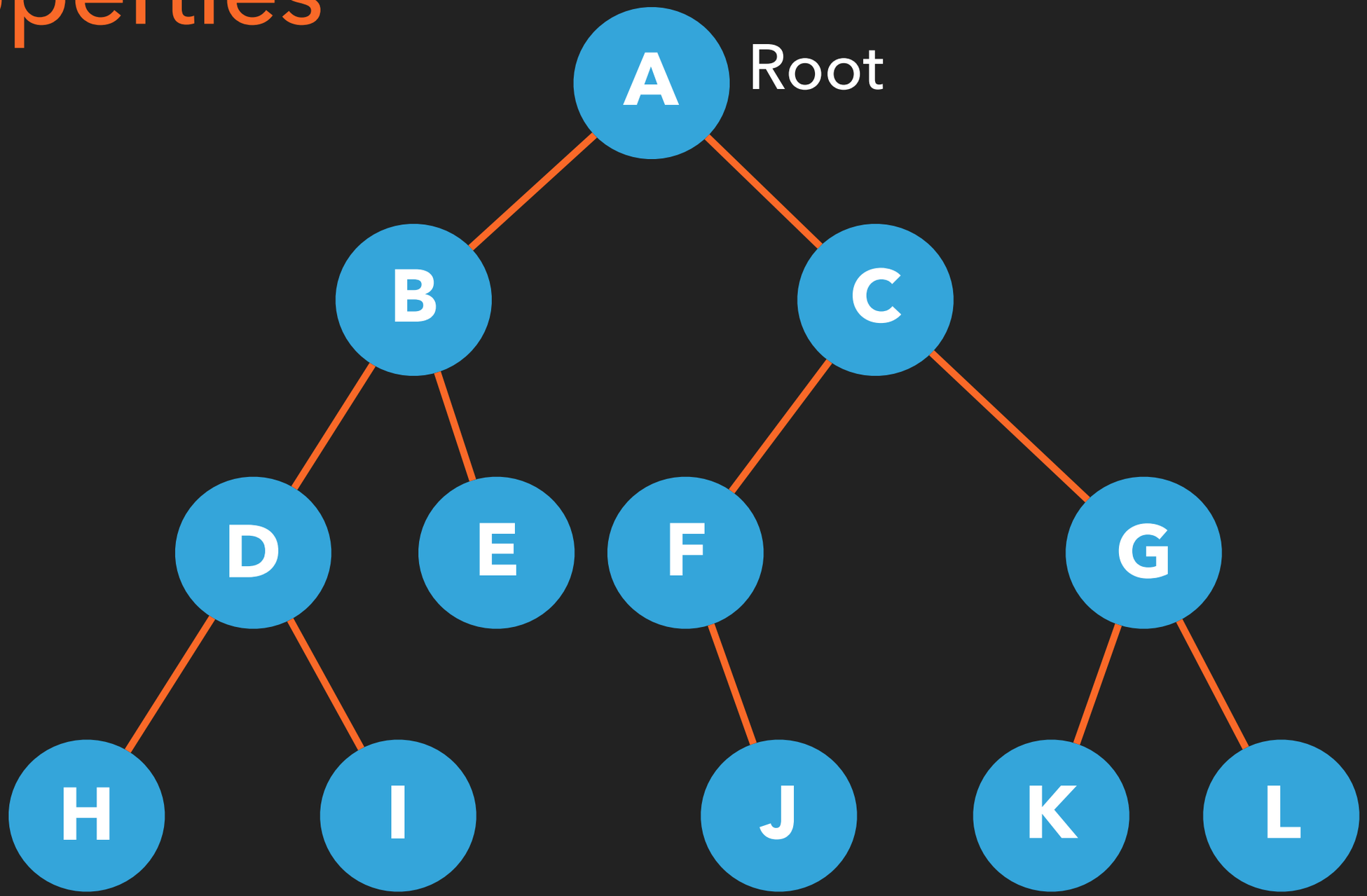


Not Complete - not full

Properties

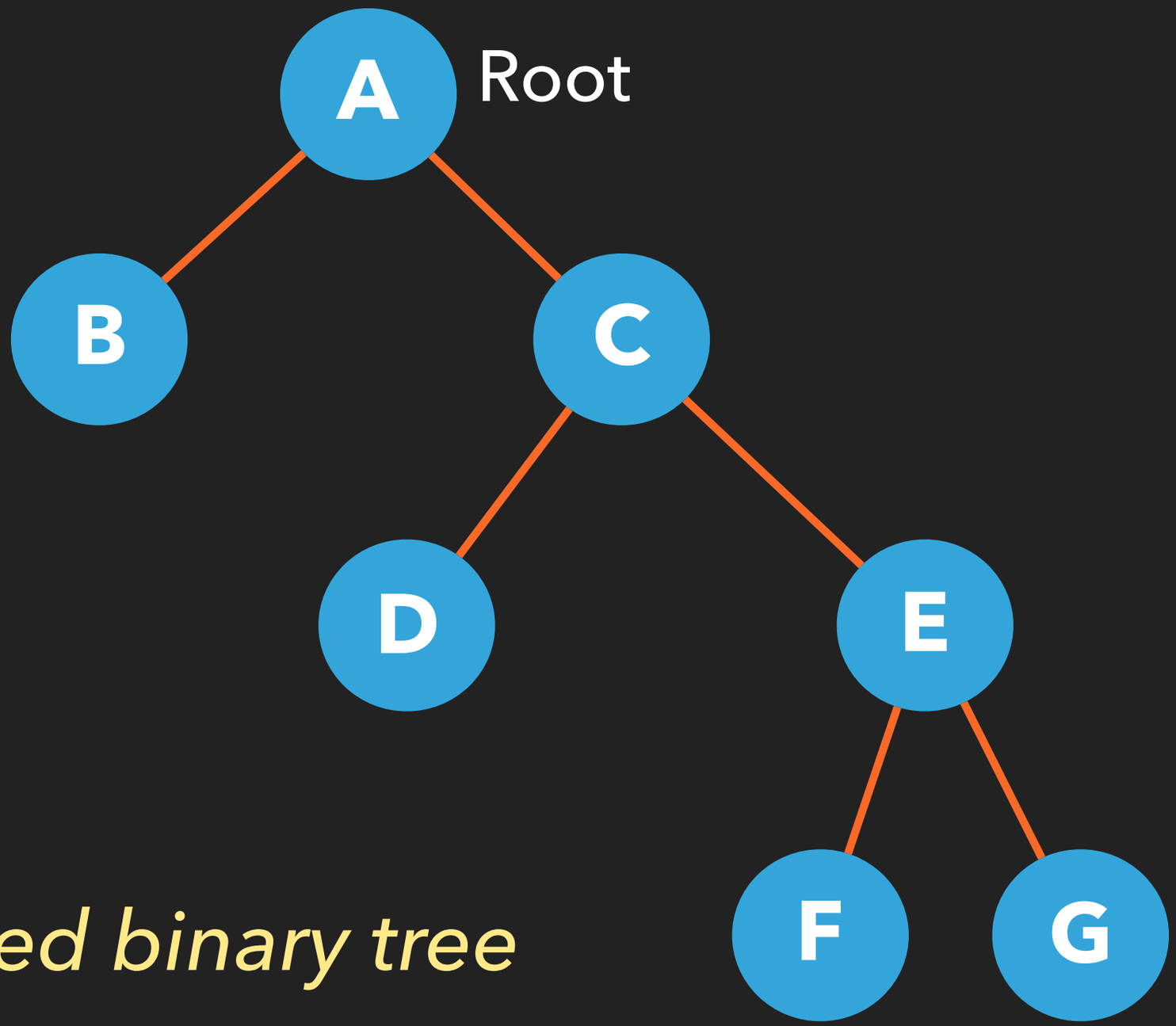
- Balanced Binary Tree: for each node, the height of the left subtree and the height of the right subtree differ by at most 1

Properties



Balanced binary tree

Properties



Unbalanced binary tree

Practice

Root?

Depth(35)?

Path from 70 to 72?

Height of the tree?

Leaf nodes?

Internal nodes?

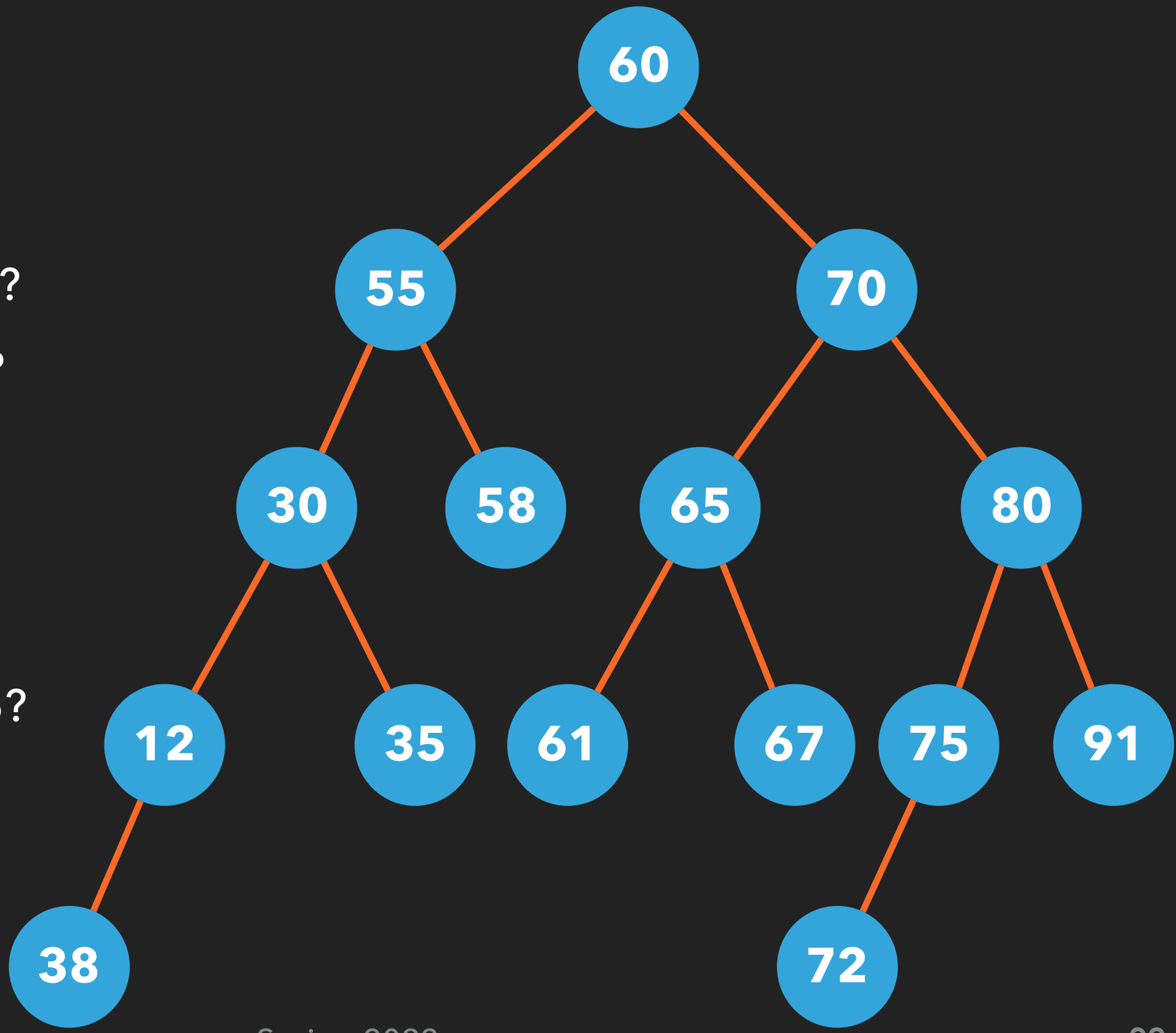
Children of 55?

Descendants of 55?

Full?

Complete?

Balanced?



Binary Tree Traversal

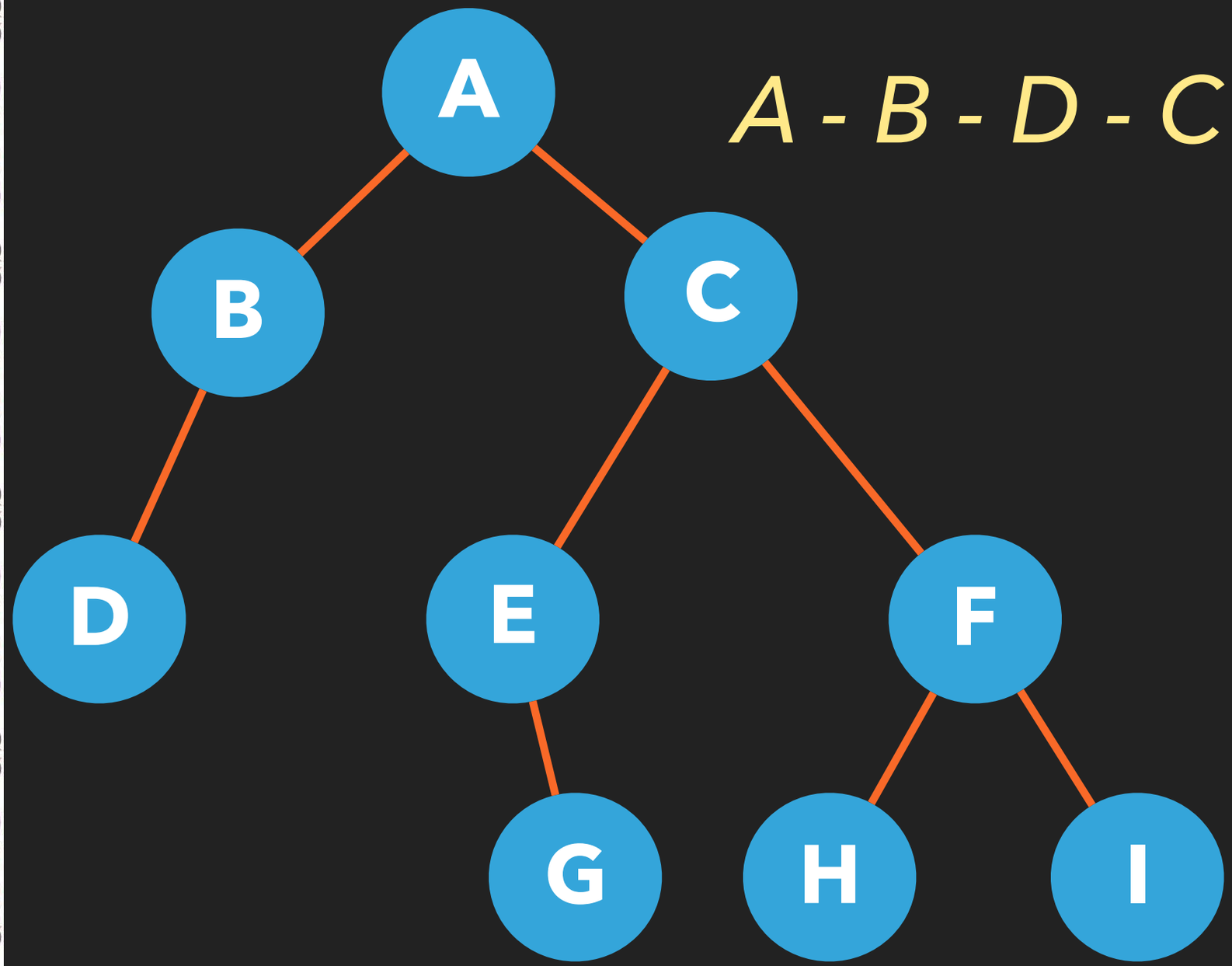
- Any process of visiting all the nodes in the tree is called traversal
- Three common traversals
  - Preorder
  - Inorder
  - Postorder

Preorder Traversal

- Any node is visited before its children
- V-L-R: Visit Node - Go Left - Go Right
- Visit the node
- Traverse the left subtree
- Traverse the right subtree

Preorder Traversal

- V-L-R: Visit Node - Go Left - Go Right



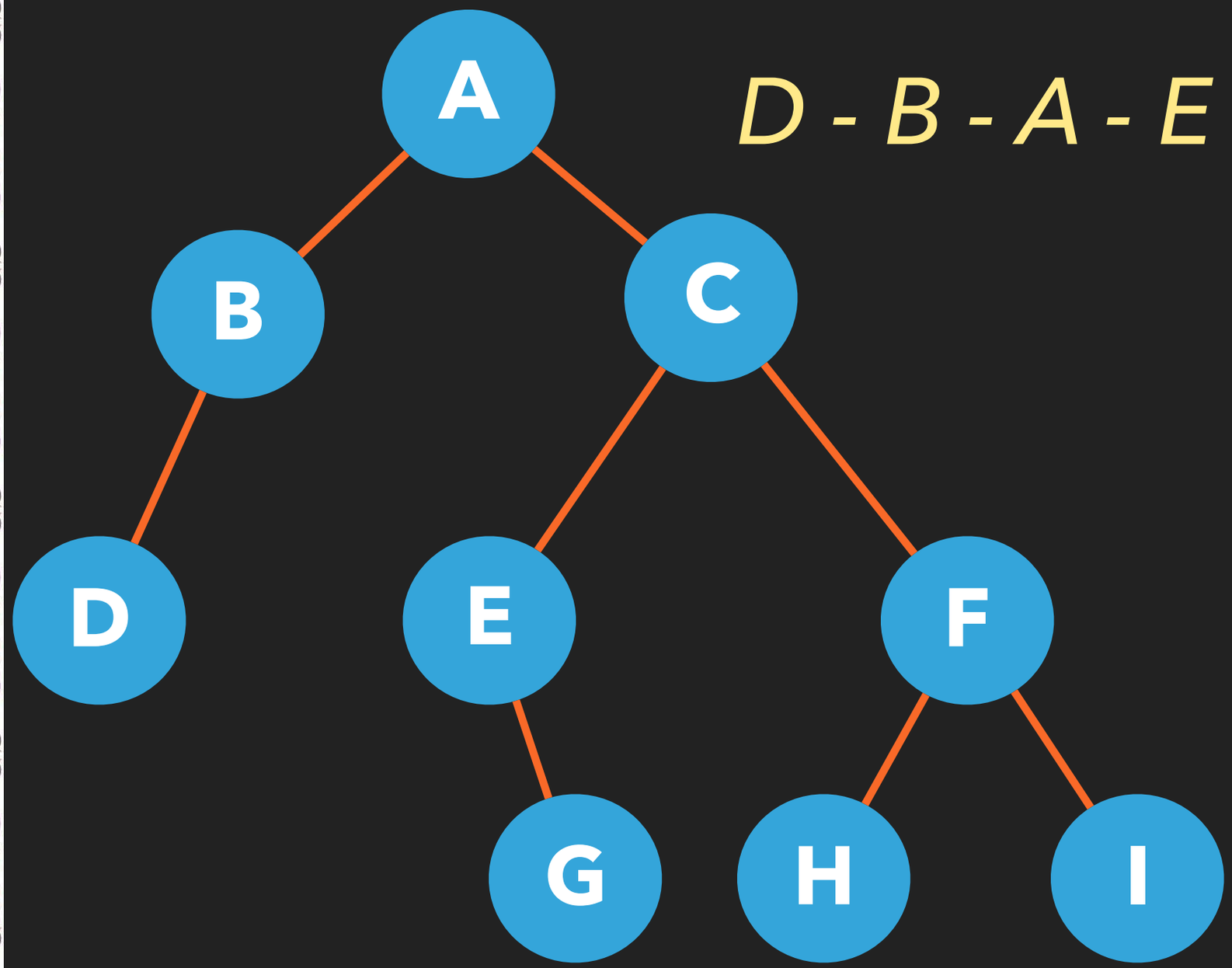
A - B - D - C - E - G - F - H - I

Inorder Traversal

- Any node is visited after its left subtree and before its right subtree
- L-V-R: Go Left - Visit Node - Go Right
  - Traverse the left subtree
  - Visit the node
  - Traverse the right subtree

In order Traversal

- L-V-R: Go Left - Visit Node - Go Right



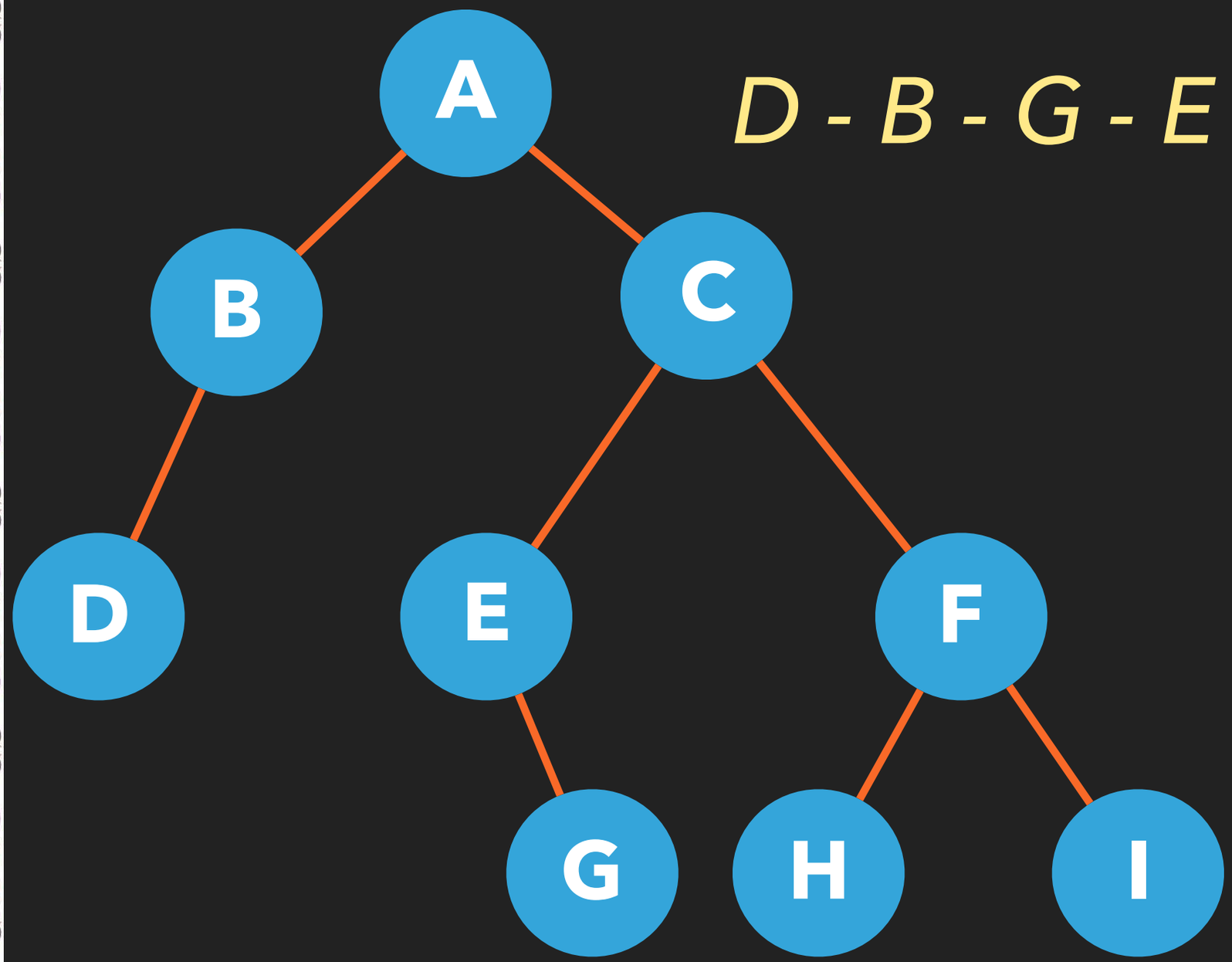
D - B - A - E - G - C - H - F - I

Postorder Traversal

- Any node is visited after its left subtree and right subtree
- L-R-V: Go Left - Go Right - Visit Node
  - Traverse the left subtree
  - Traverse the right subtree
  - Visit the node

Postorder Traversal

- L-R-V: Go Left - Go Right - Visit Node



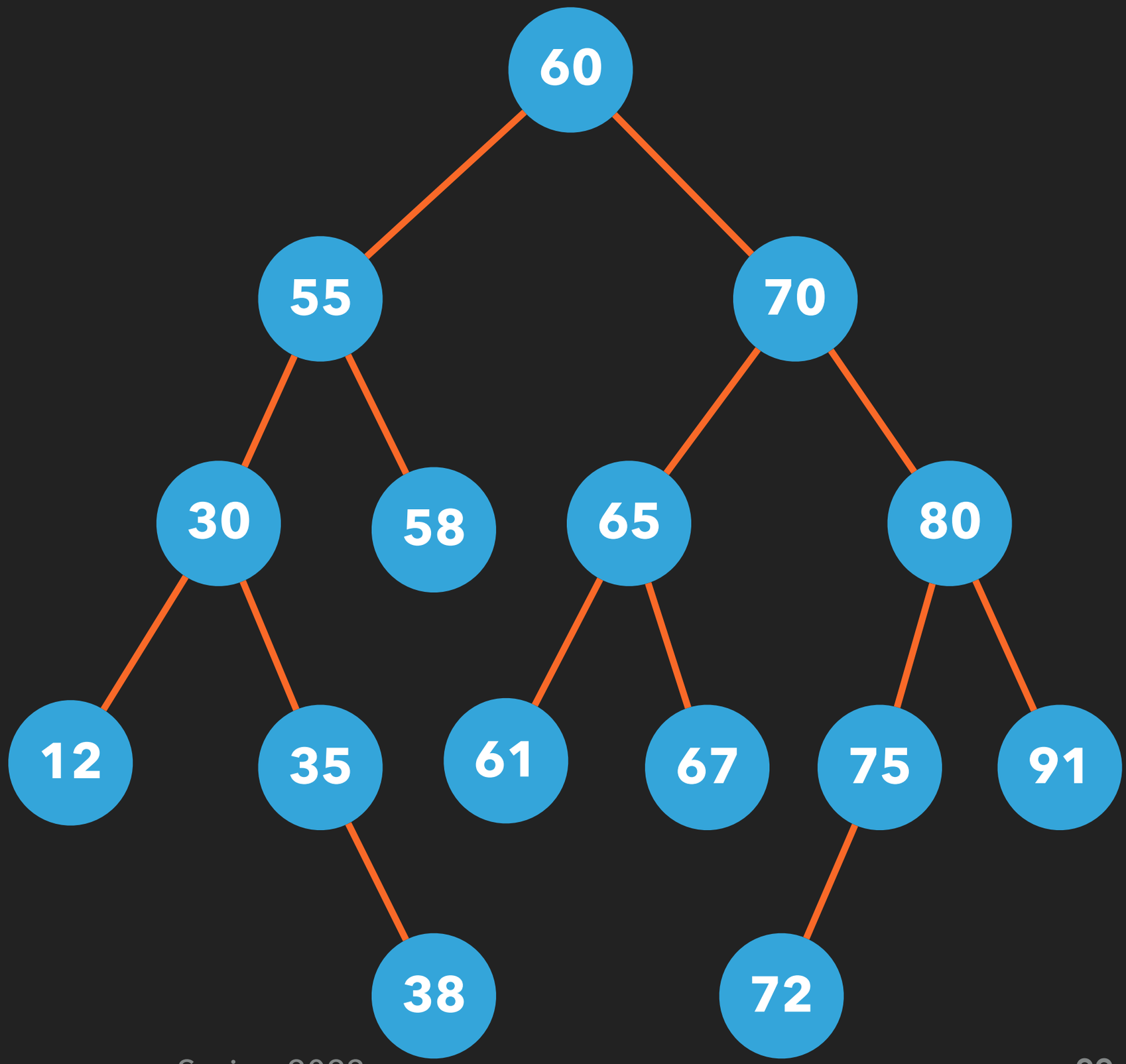
D - B - G - E - H - I - F - C - A

Practice

Preorder: ?

Inorder: ?

Postorder: ?



Next

- Special binary trees
  - Heap
  - Binary Search Tree (BST)