

# Trees

February 9



# What is a *binary* tree?

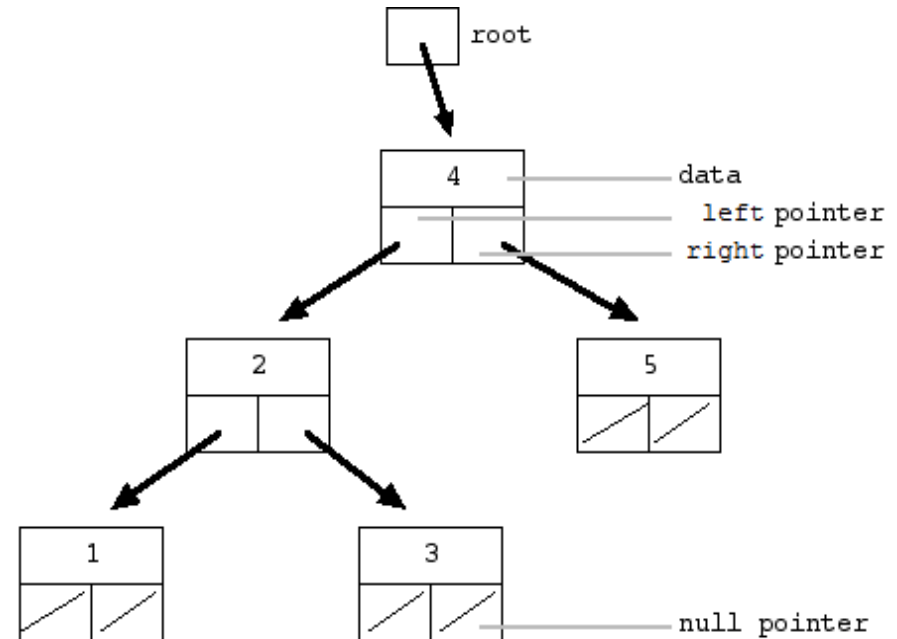


# What is a *binary* tree in CS?



# What is a *binary* tree ?

- A set of *nodes linked* together.
- Each node consists of *data* and *links* to two *children*.
- The *root* points to the *first* node of the tree.



- Which nodes are the *children* of 4?
- How is tree different from a *binary* tree?

•

•

# Traversing a Tree

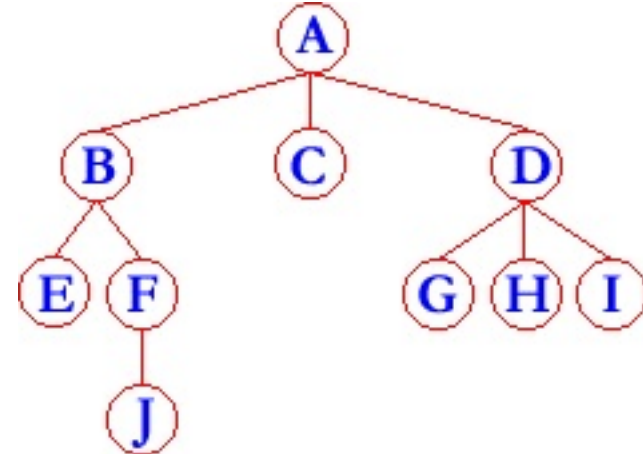
Q. What does *traverse* mean?

A. To *walk* or *travel* from node to node.

Pre-order Traversal:

We visit a node, then it's children from left to right.

A B E F J C D G H I



# Traversing a Tree

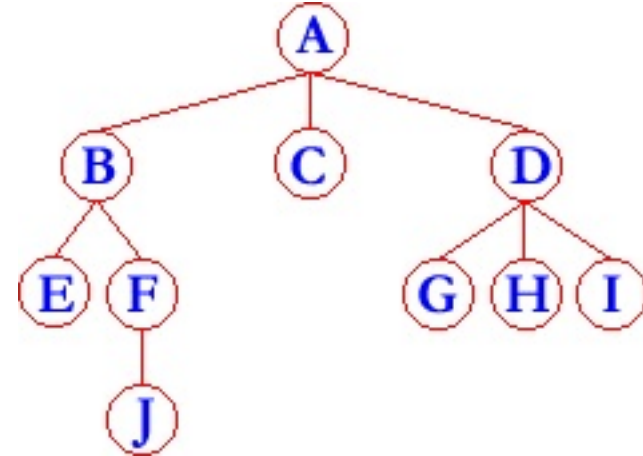
Q. What does *traverse* mean?

A. To *walk* or *travel* from node to node.

Post-order Traversal:

We visit a node's children from left to right and then the node.

E J F B C G H I D A



Q. When would you use each of these orderings?

•

•

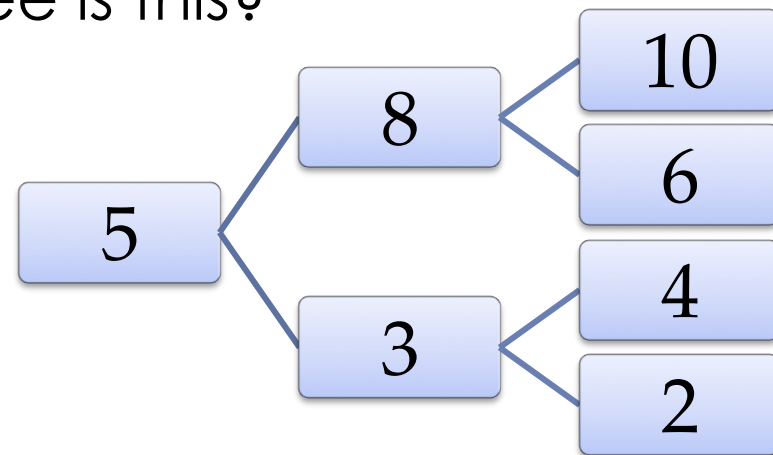
# Special Binary Trees

Q. What do you notice about the *children* of each node?

A. The *left child* is *less* than the node. The *right child* is *greater* than the node.

Q. What *type* of binary tree is this?

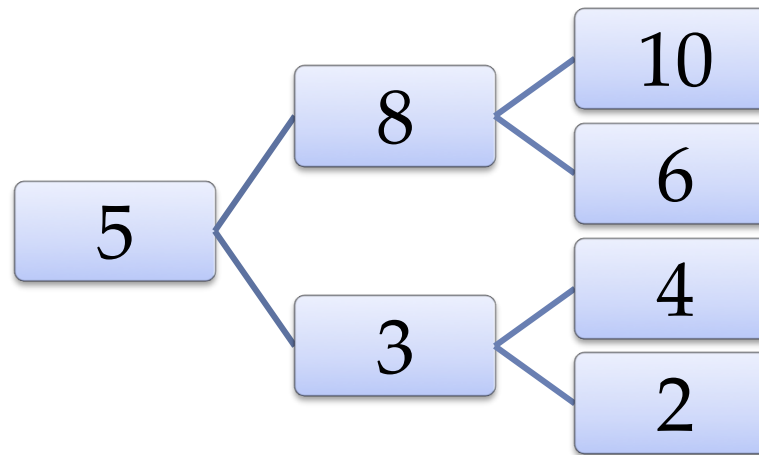
A. A binary *search* tree.



# Traversing a BST

In-order Traversal:

We visit a node's left child, then the node, then the right child



What is the ordering?

2 3 4 5 6 8 10

•

•