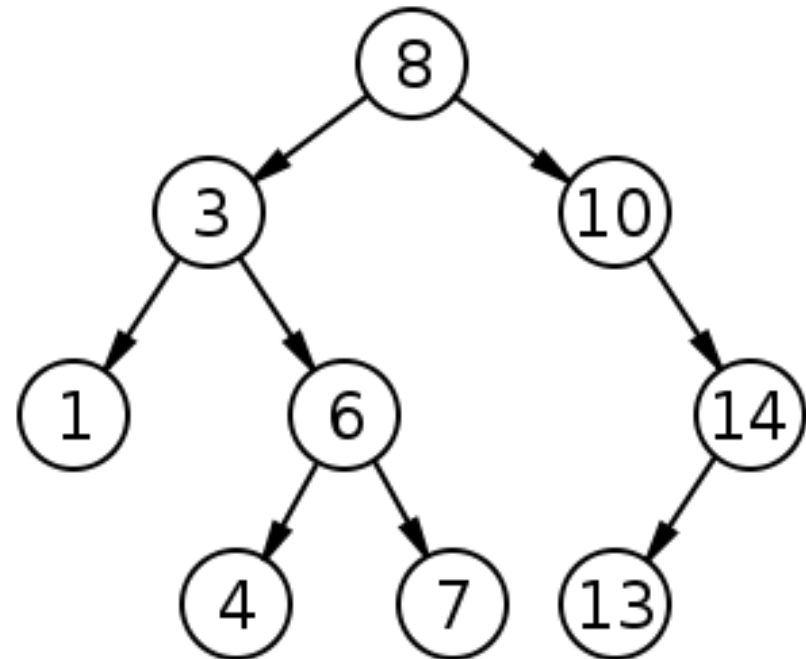


# CS 112 - Fall 2012, Lab 08

Haohan Zhu

# Tree - Data Structure

- Basic Concept:
  - Root
  - Leaf
  - Parent (Ancestor)
  - Child
  - Siblings
  - Height / Depth



# Several Trees

- Binary Tree
- Binary Search Tree
- Binary Expression Tree
- B-tree
- Huffman Tree (Information Theory)
- Red-black Tree
- BSP tree, R-tree, k-d tree(Space Partition)
- ...

# Tree Traversal

- Depth-first traversal
  - preorder
  - inorder
  - postorder
- Breadth-first traversal
  - level-order

# Tree Traversal

- Preorder

*preorder(node)*

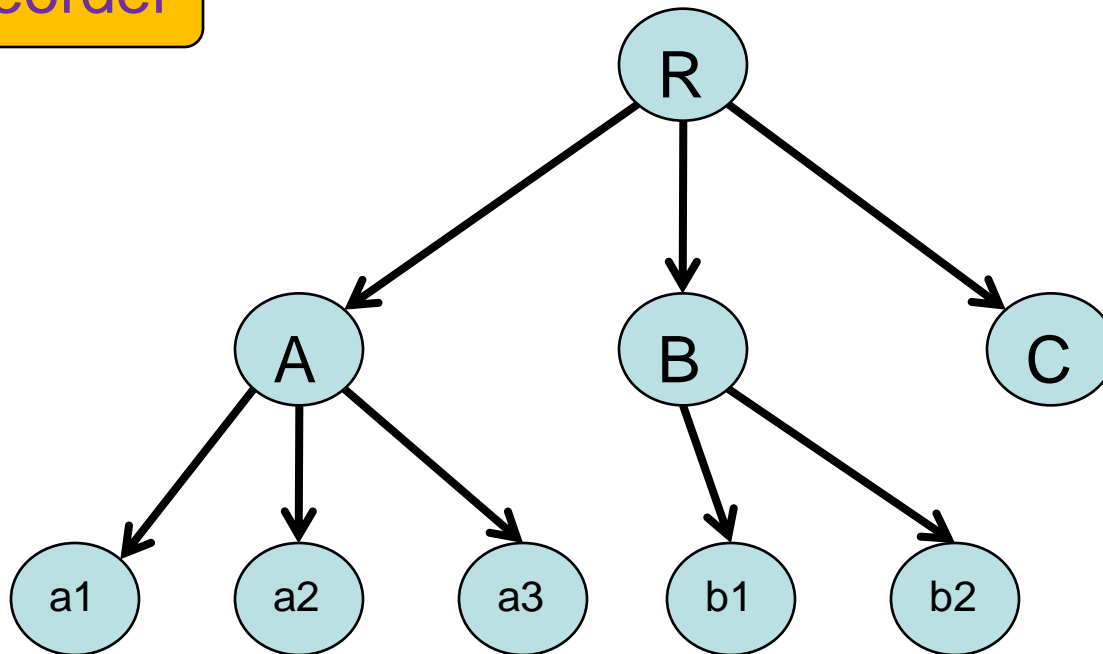
*if node == null then return*

*visit(node)*

*preorder(node.children)*

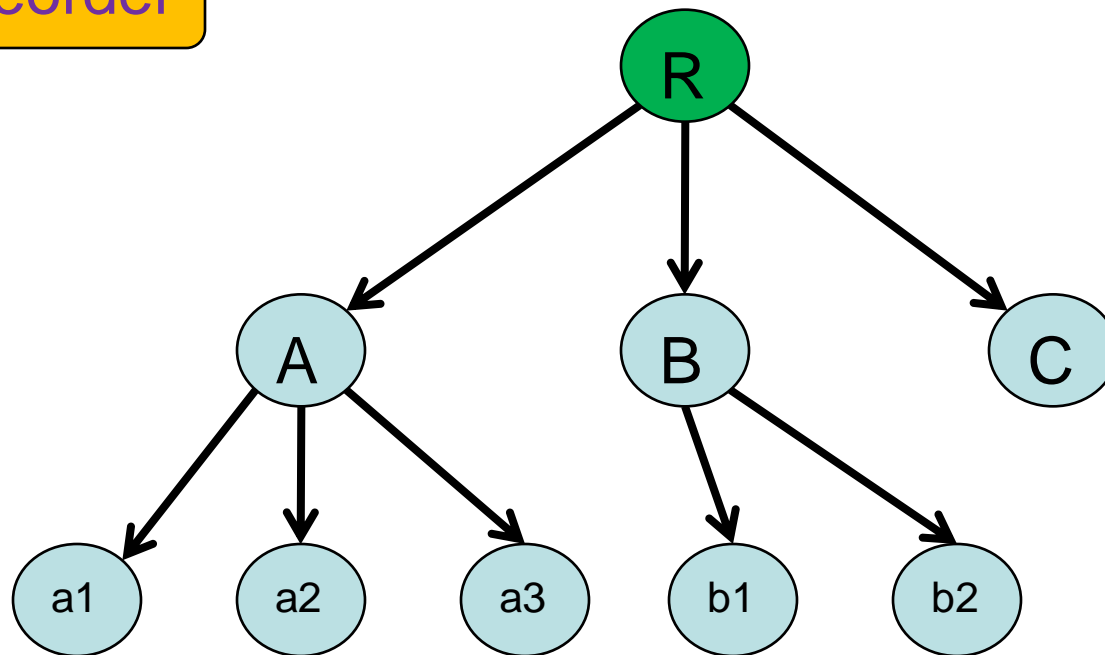
# Tree Traversal

Preorder



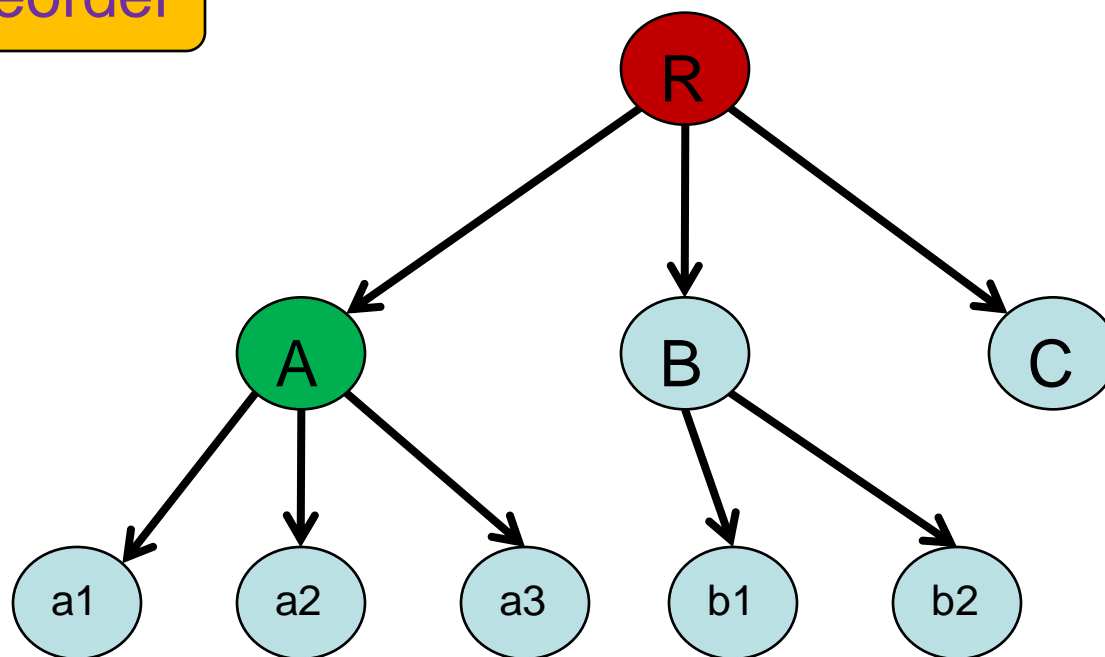
# Tree Traversal

Preorder



# Tree Traversal

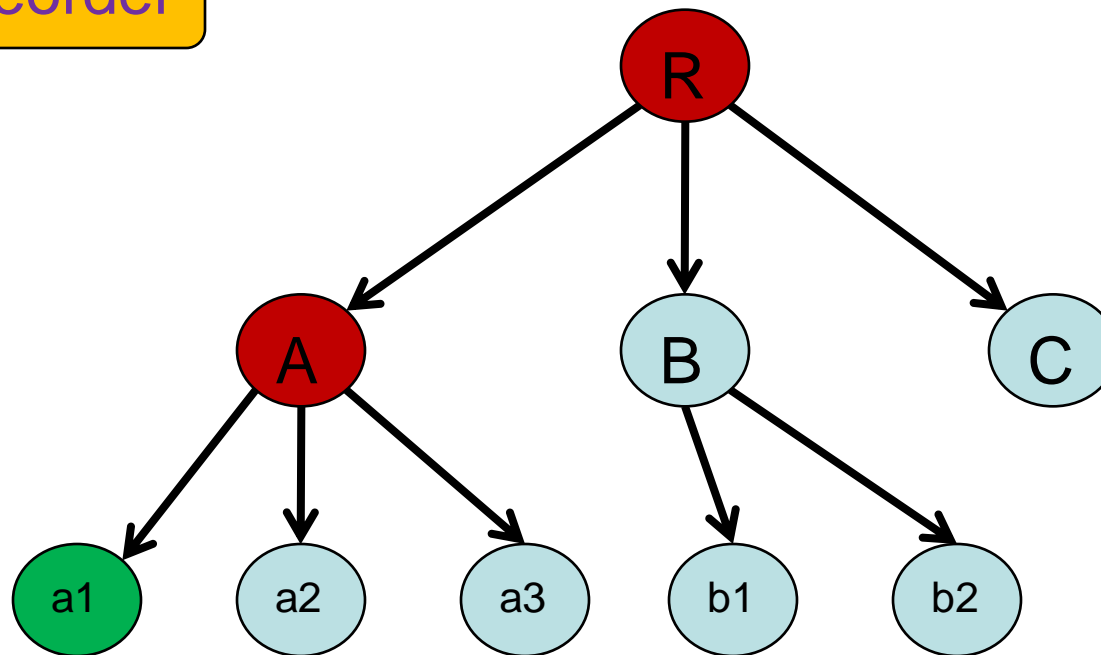
Preorder





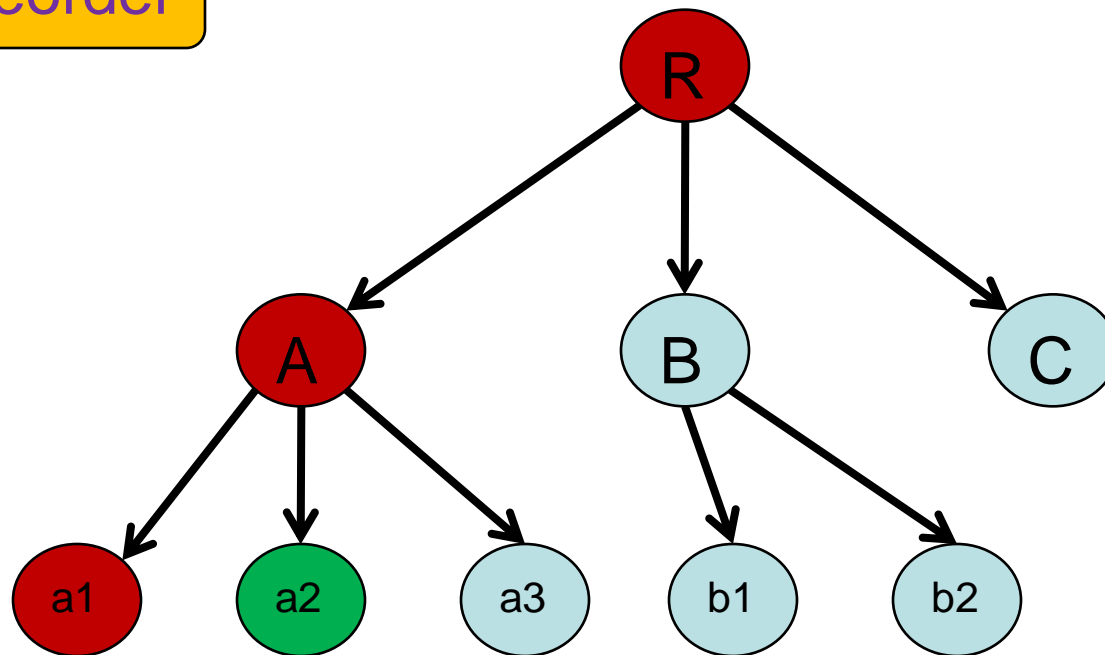
# Tree Traversal

Preorder



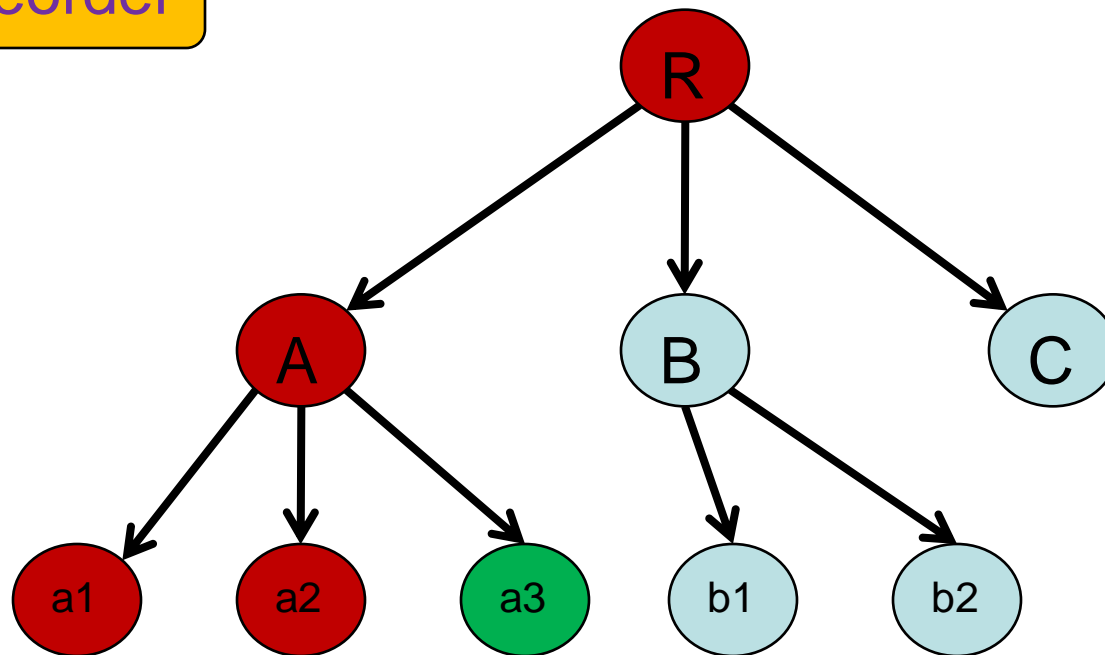
# Tree Traversal

Preorder



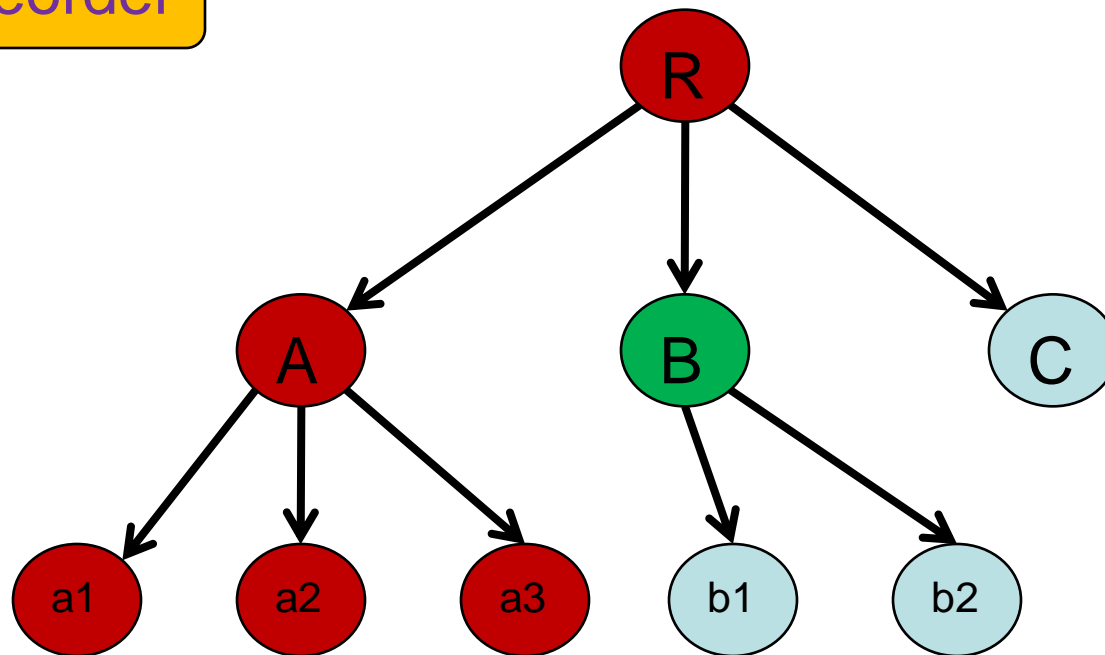
# Tree Traversal

Preorder



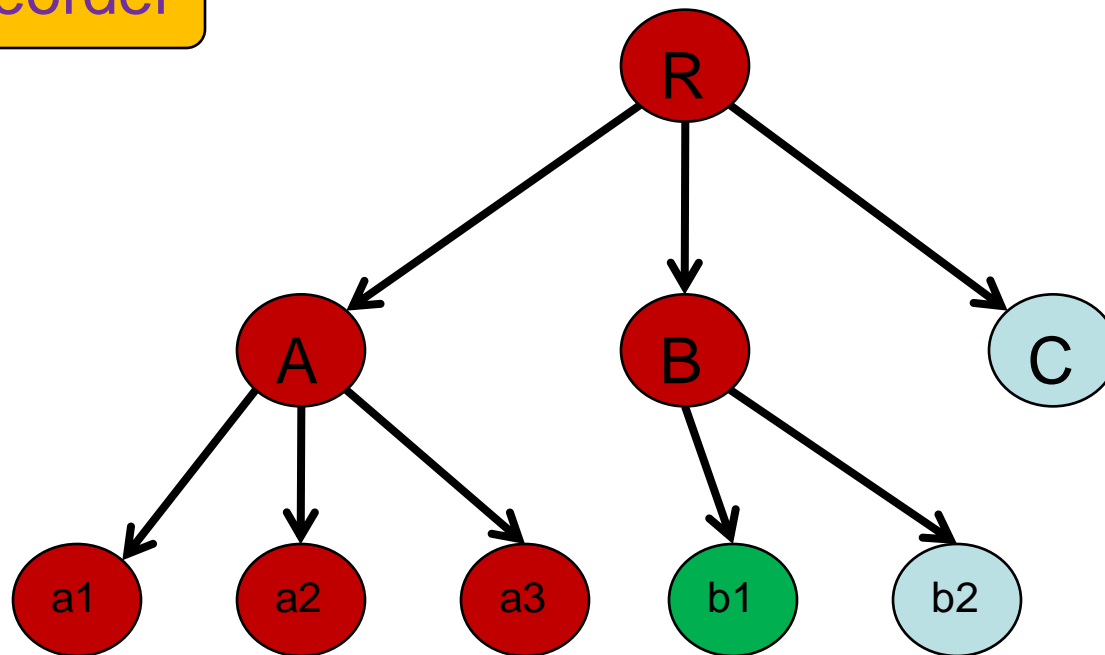
# Tree Traversal

Preorder



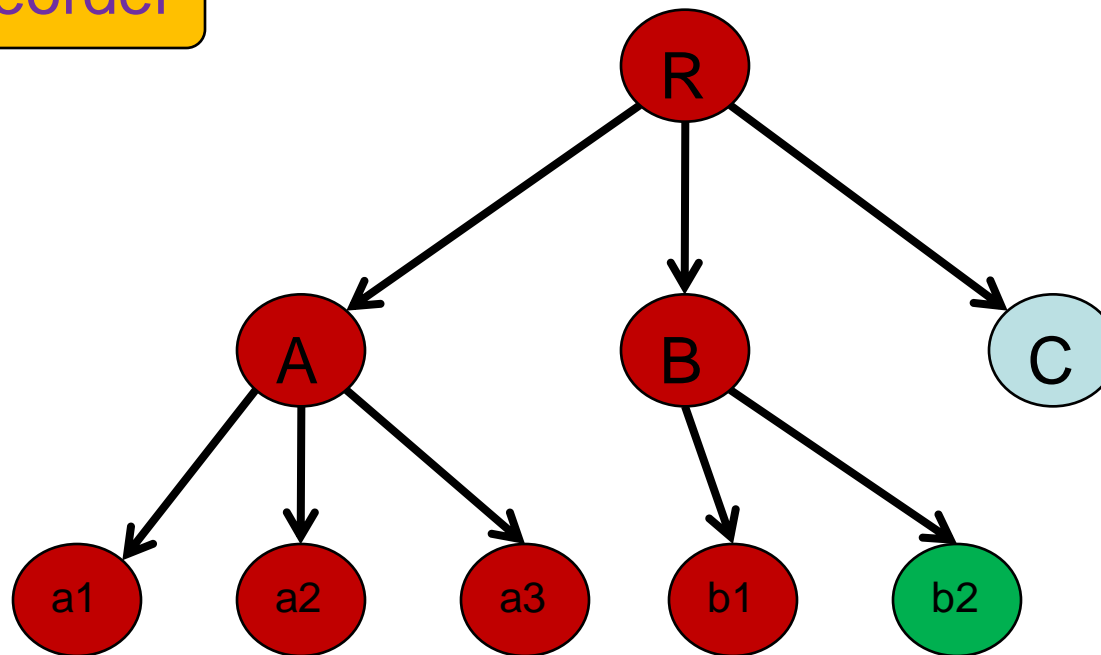
# Tree Traversal

Preorder



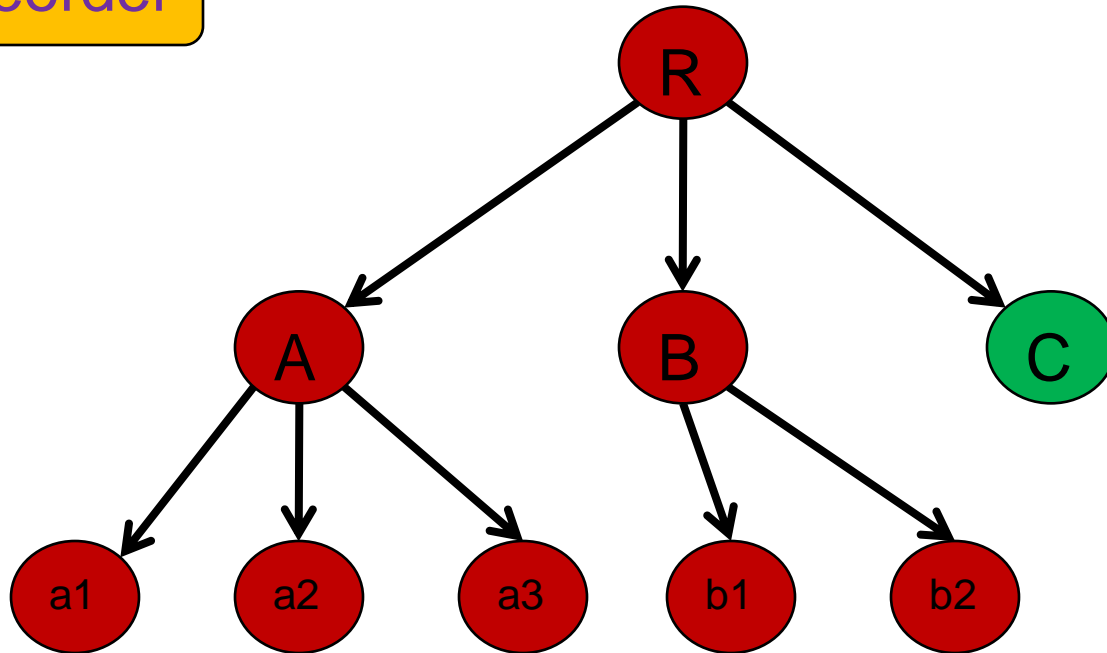
# Tree Traversal

Preorder



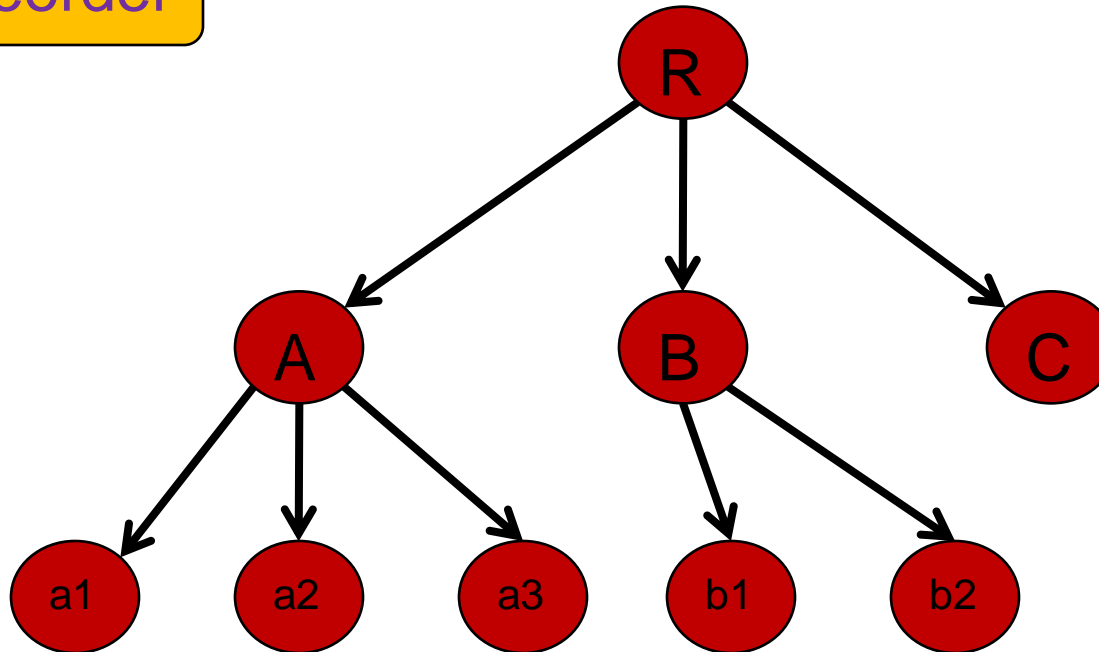
# Tree Traversal

Preorder



# Tree Traversal

Preorder





# Tree Traversal

- Postorder

*postorder(node)*

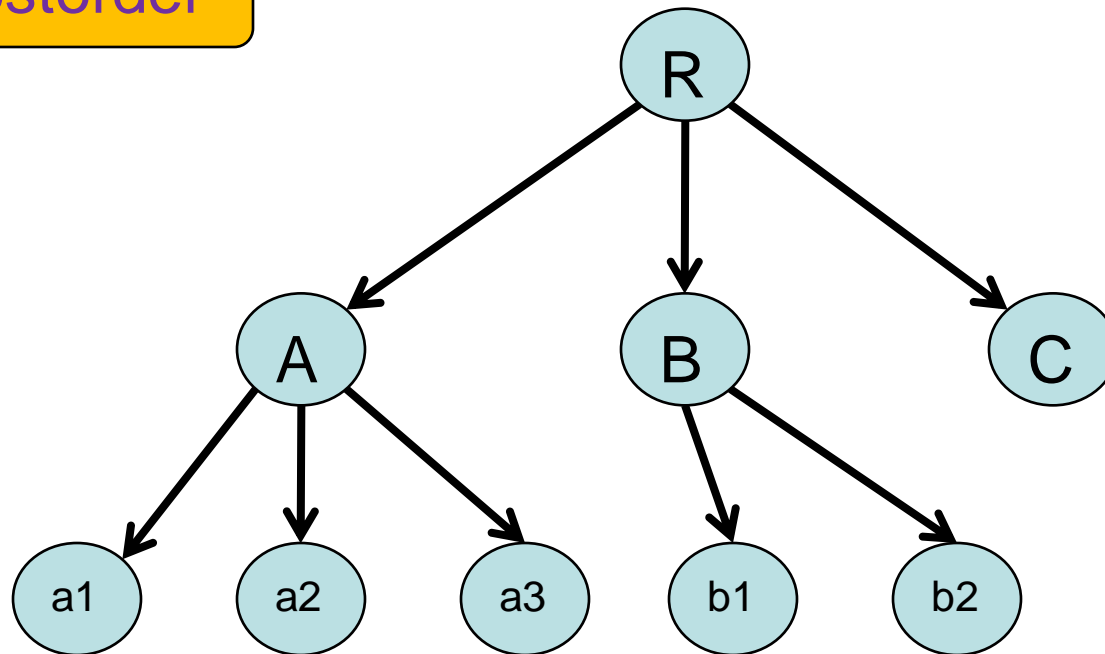
*if node == null then return*

*postorder(node.children)*

*visit(node)*

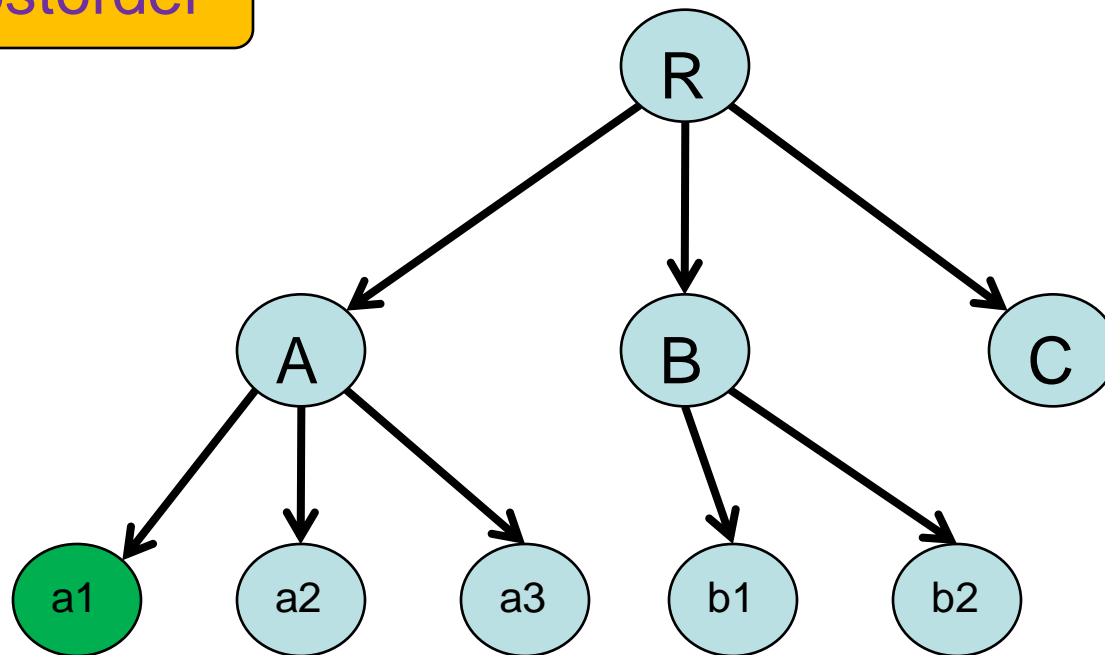
# Tree Traversal

Postorder



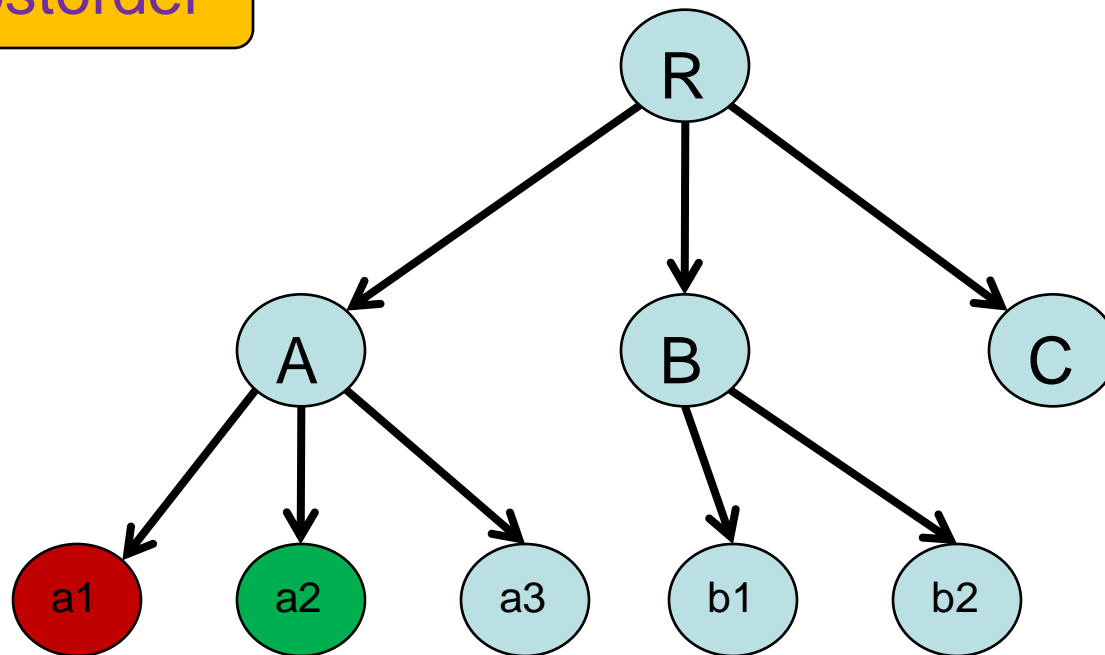
# Tree Traversal

Postorder



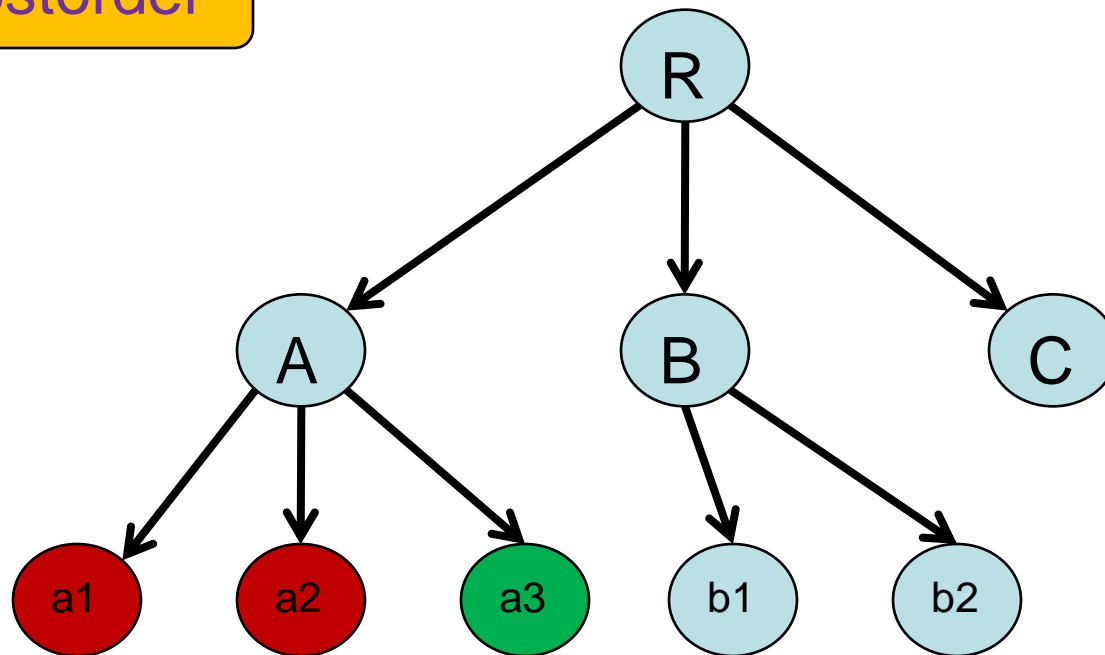
# Tree Traversal

Postorder



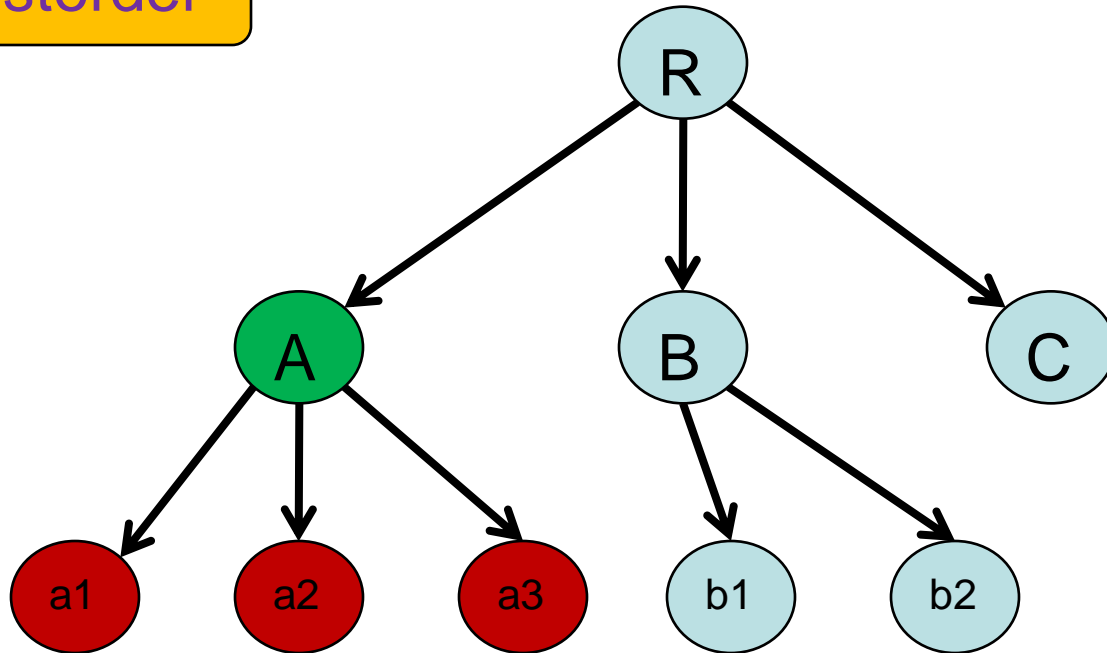
# Tree Traversal

Postorder



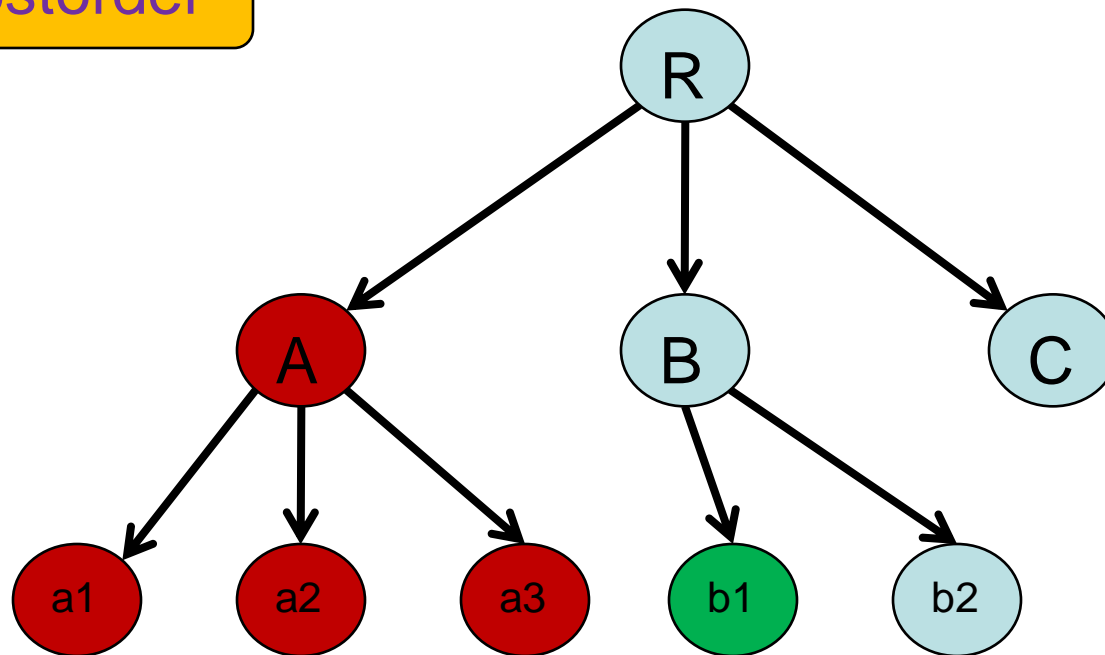
# Tree Traversal

Postorder



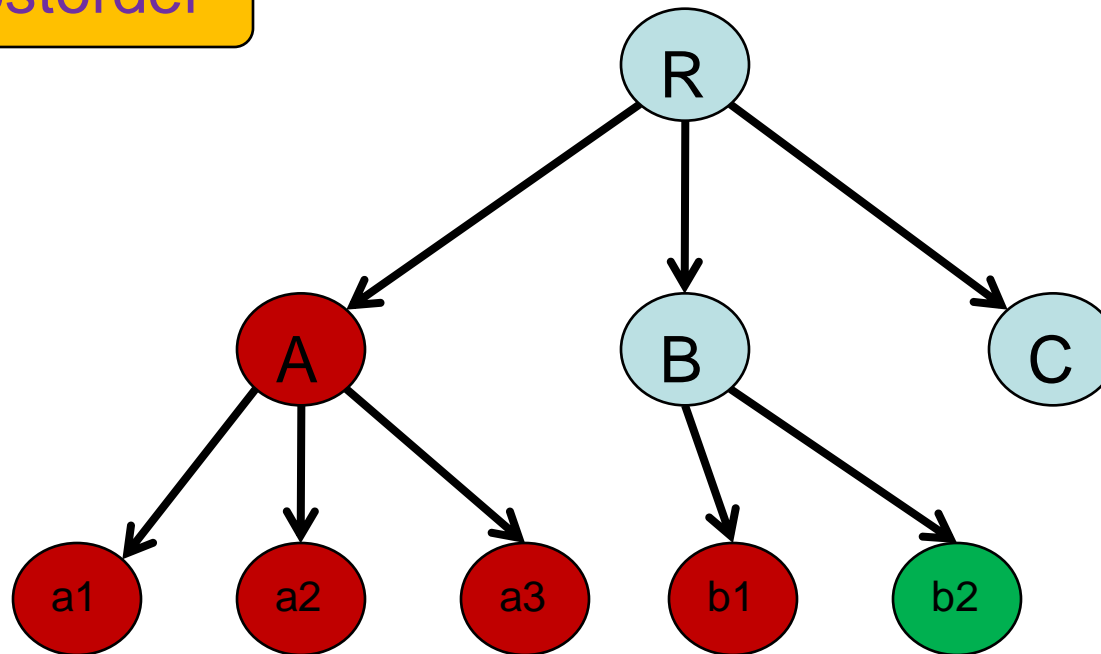
# Tree Traversal

Postorder



# Tree Traversal

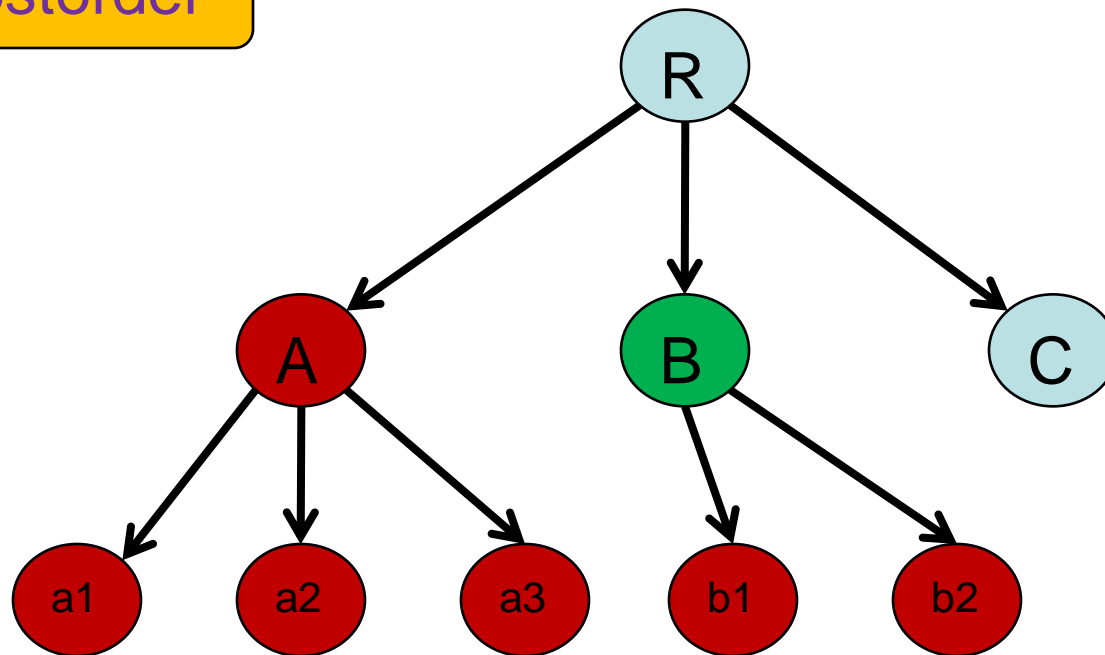
Postorder





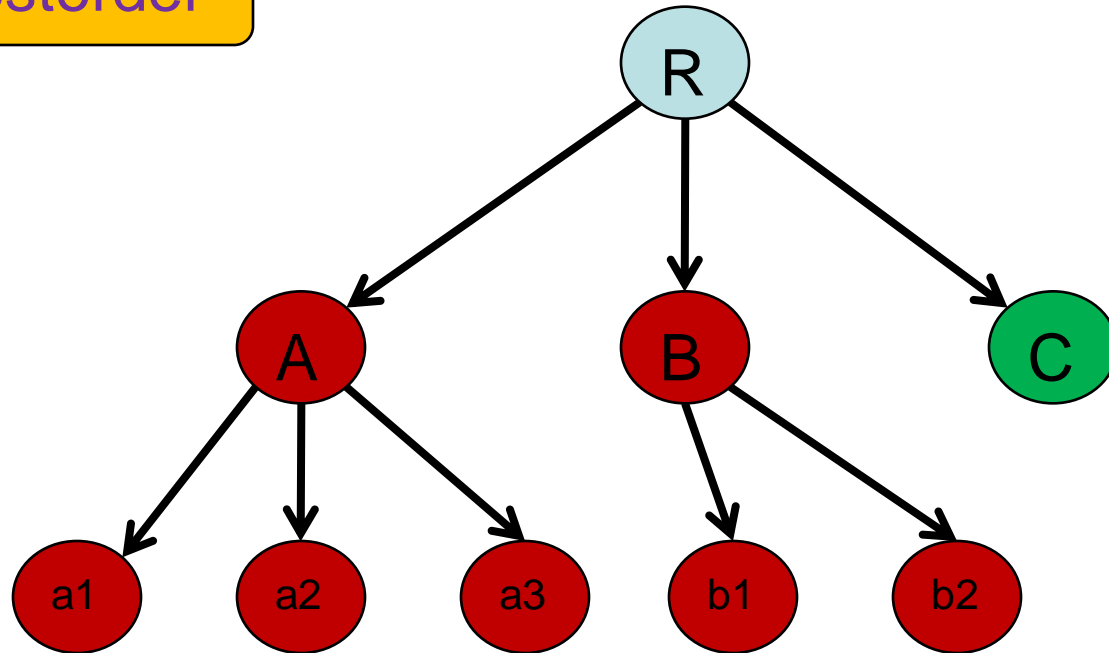
# Tree Traversal

Postorder



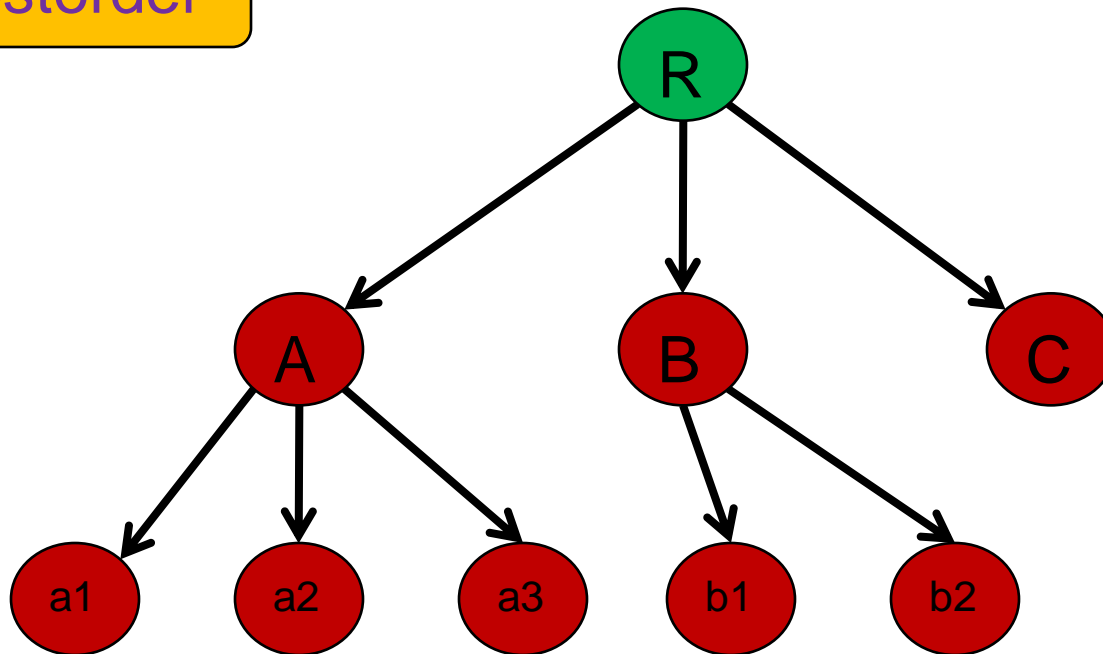
# Tree Traversal

Postorder



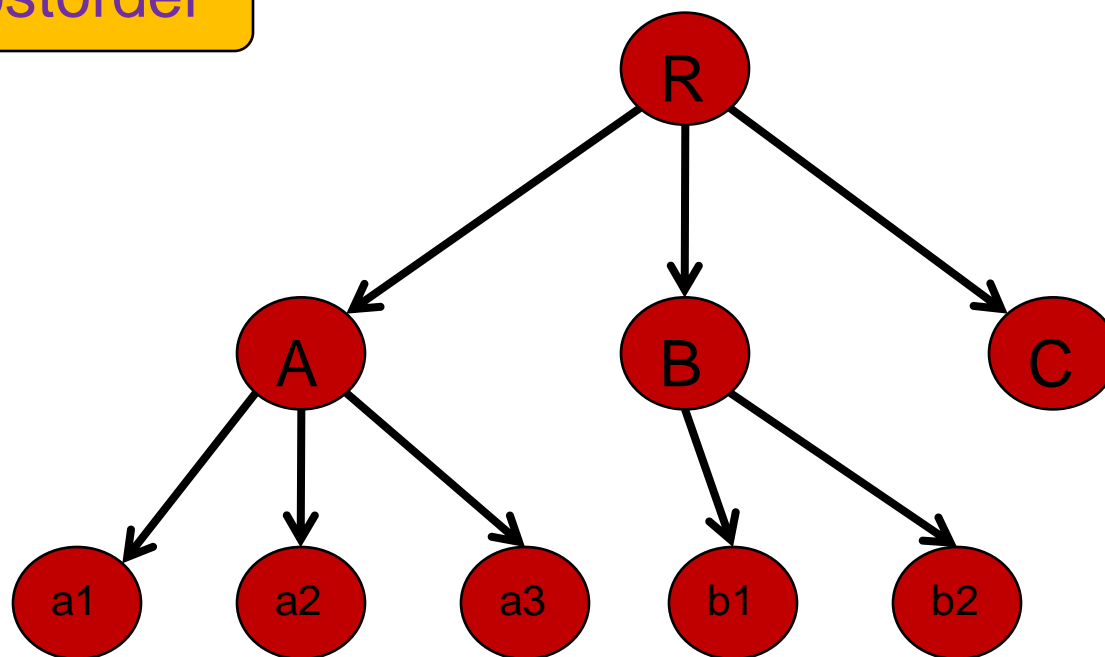
# Tree Traversal

Postorder



# Tree Traversal

Postorder



# Tree Traversal

- Inorder

*inorder(node)*

*if node == null then return*

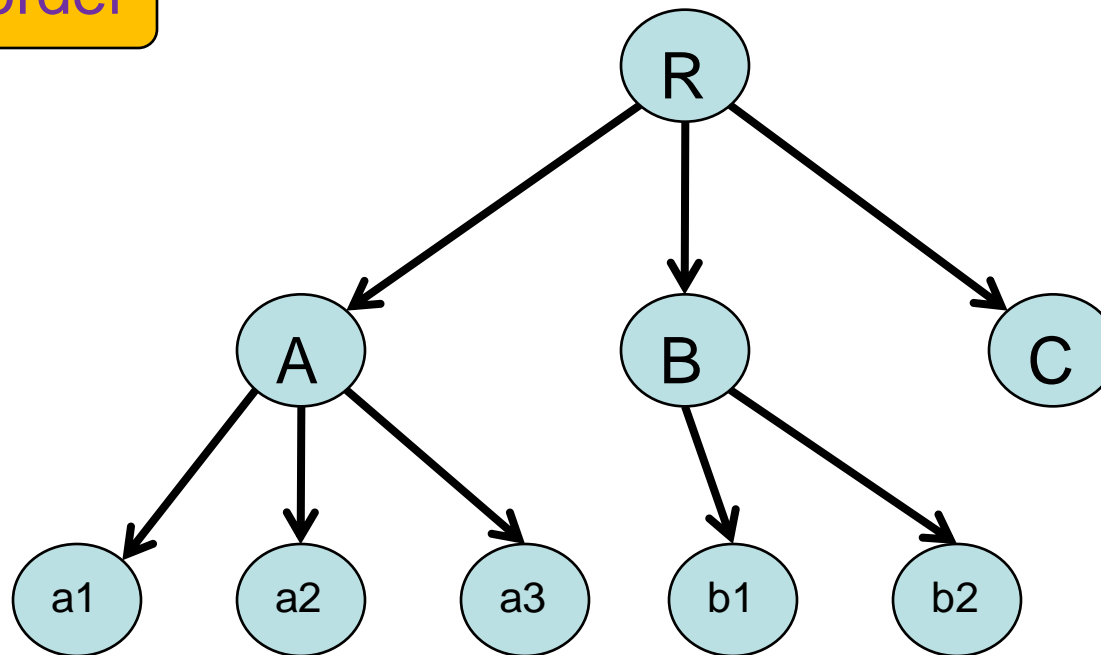
*inorder(node.first\_child or node.left)*

*visit(node)*

*inorder(node.other\_children or node.right)*

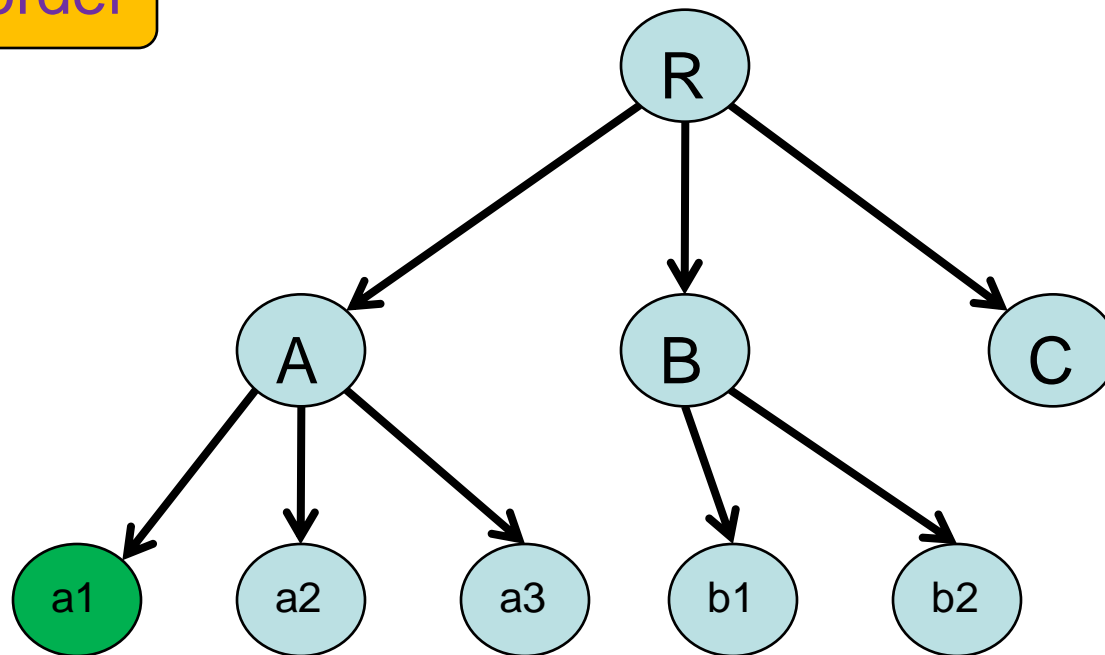
# Tree Traversal

Inorder



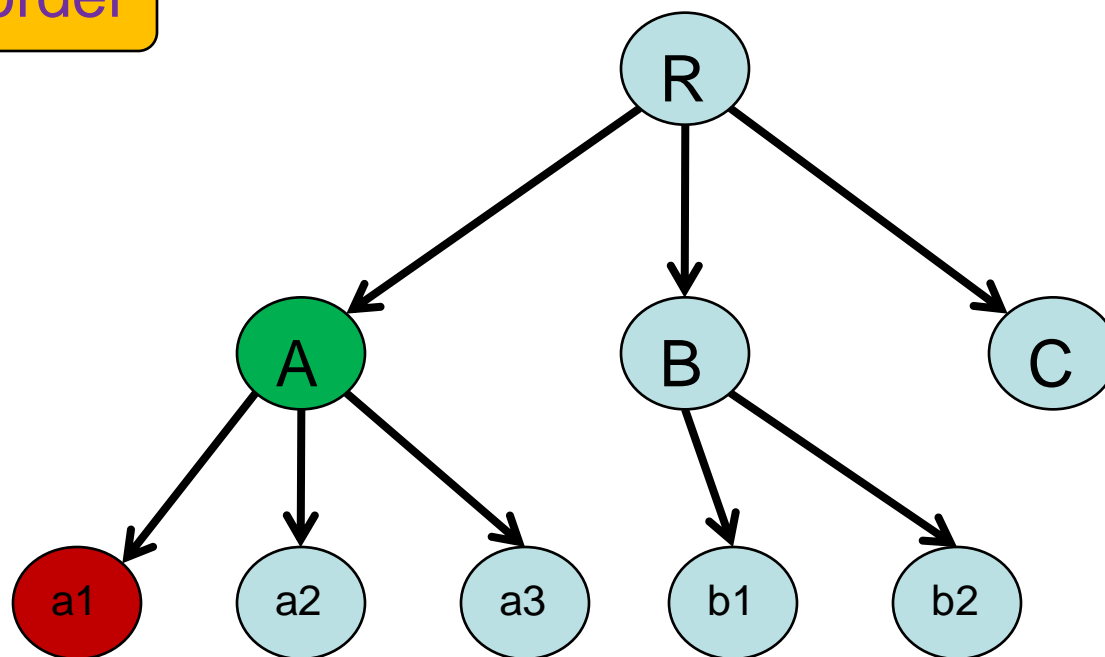
# Tree Traversal

Inorder



# Tree Traversal

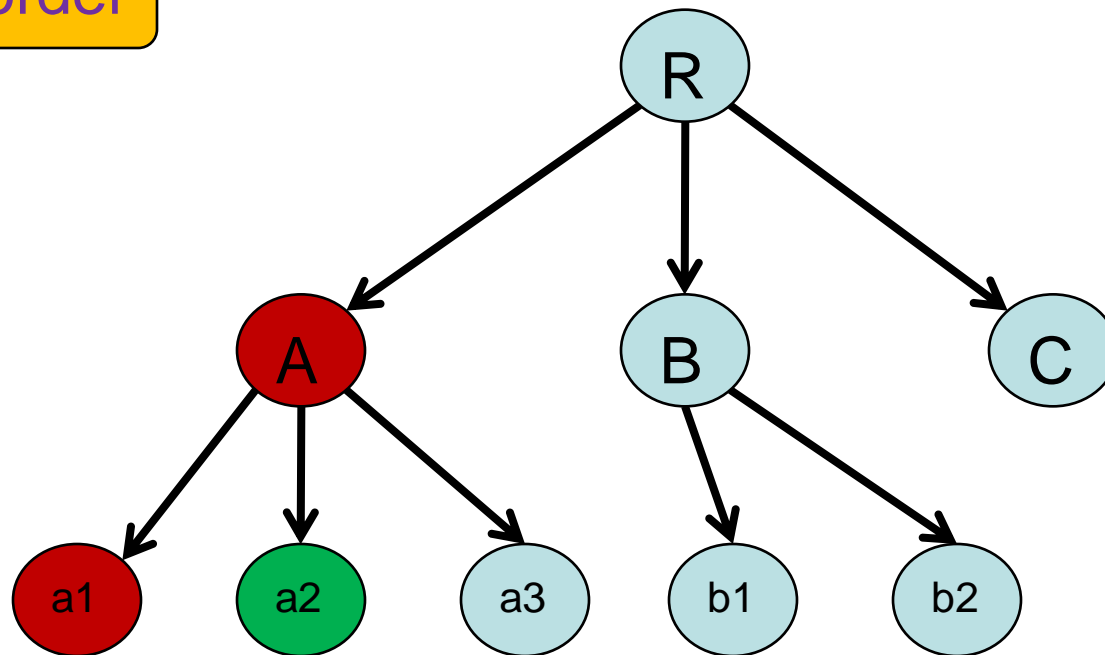
Inorder





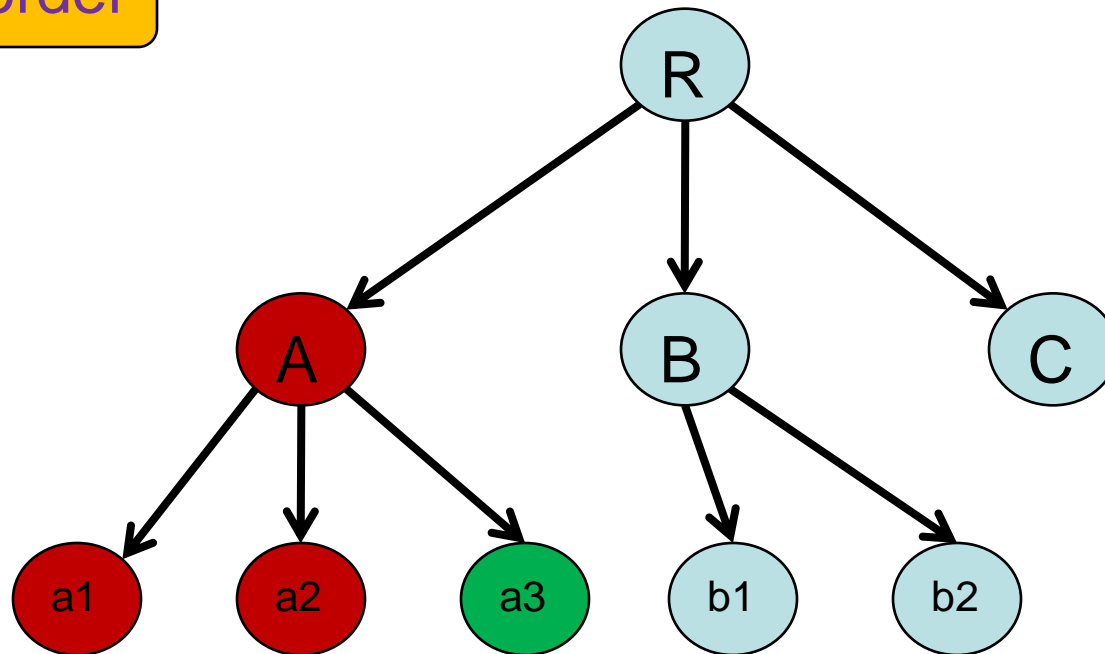
# Tree Traversal

Inorder



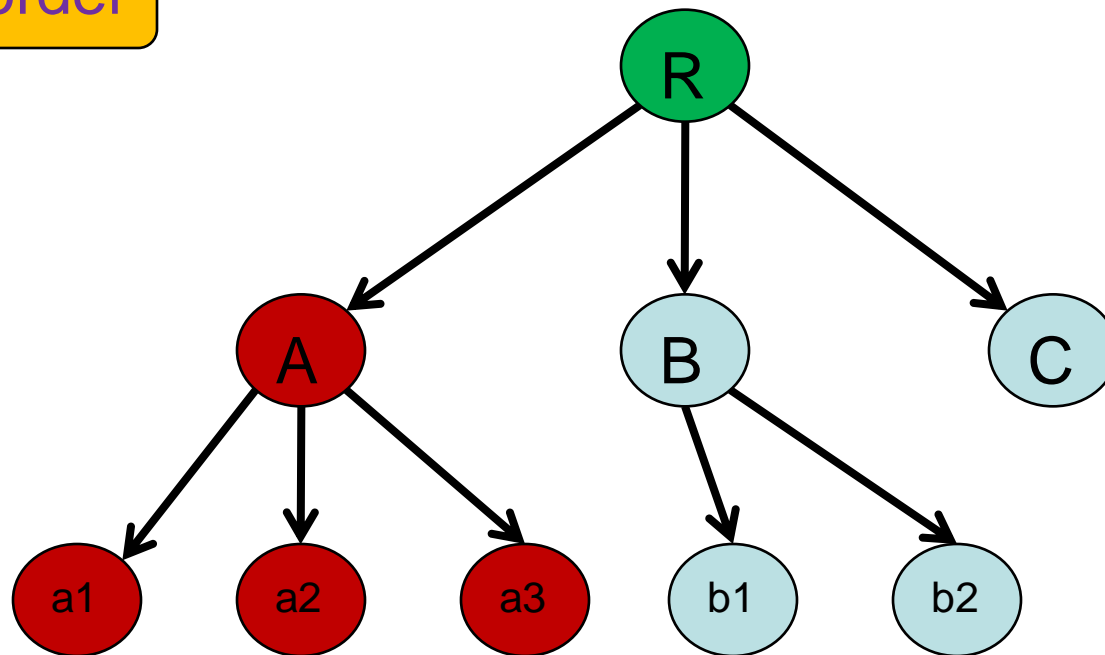
# Tree Traversal

Inorder



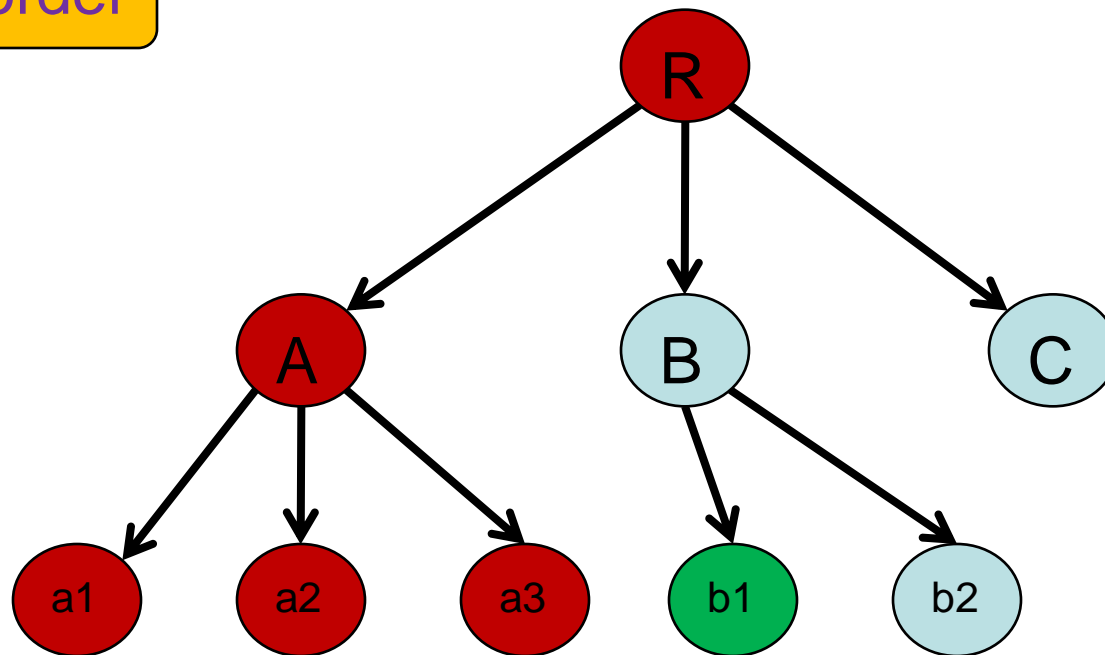
# Tree Traversal

Inorder



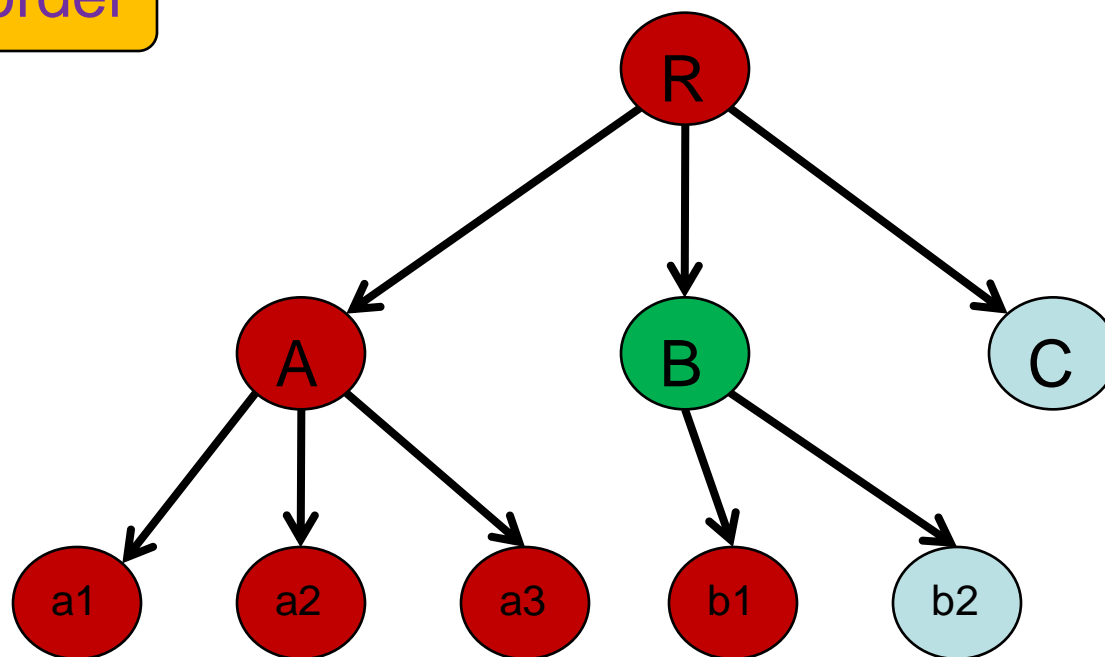
# Tree Traversal

Inorder



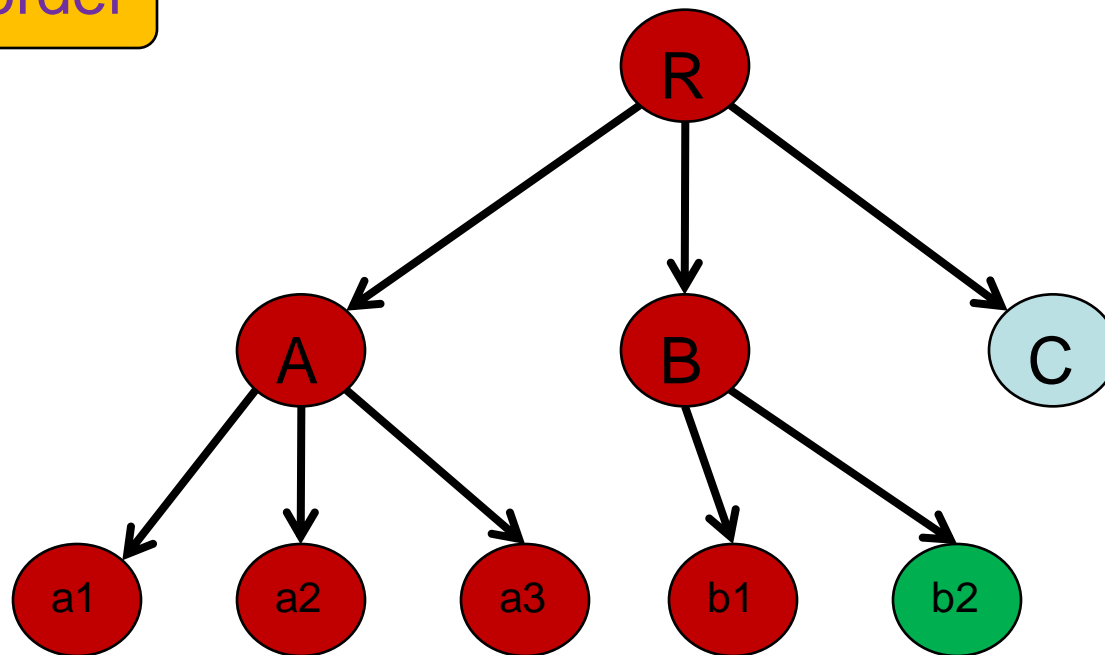
# Tree Traversal

Inorder



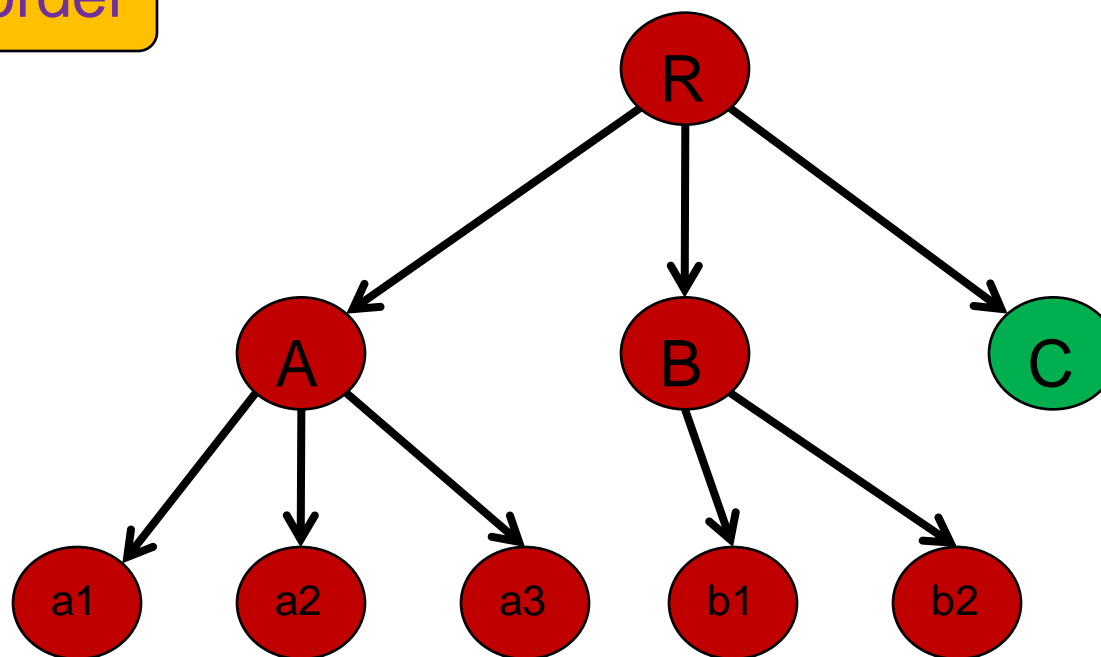
# Tree Traversal

Inorder



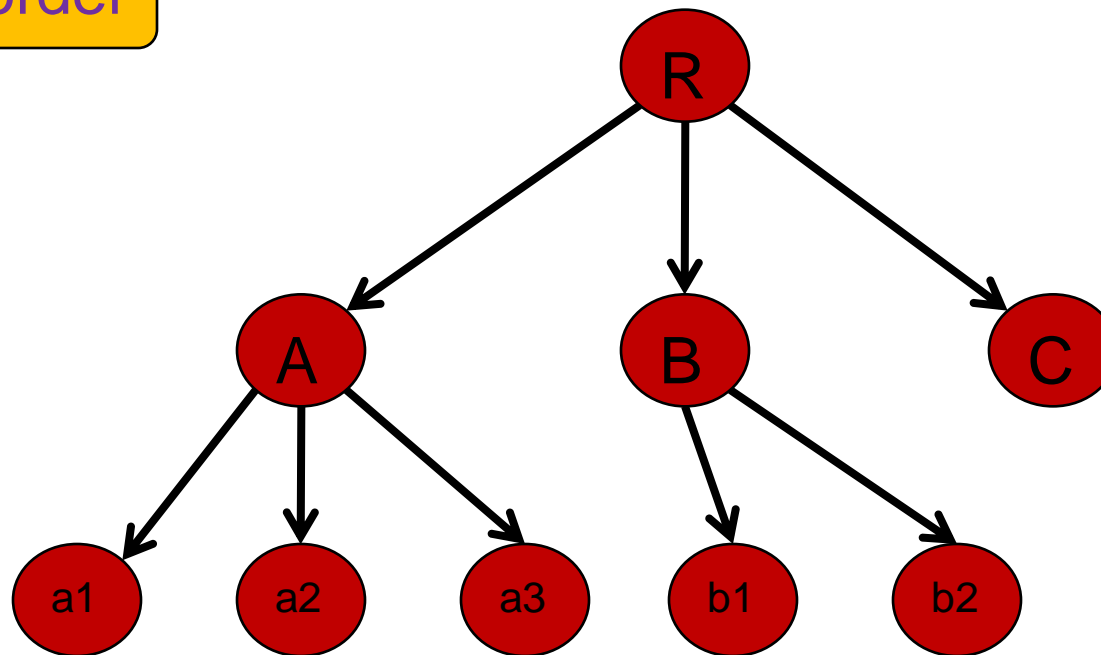
# Tree Traversal

Inorder



# Tree Traversal

Inorder





# Practice

- Write Inorder traversing algorithm