SCNU - UOA
OBJECT-ORIENTED PROGRAMMING

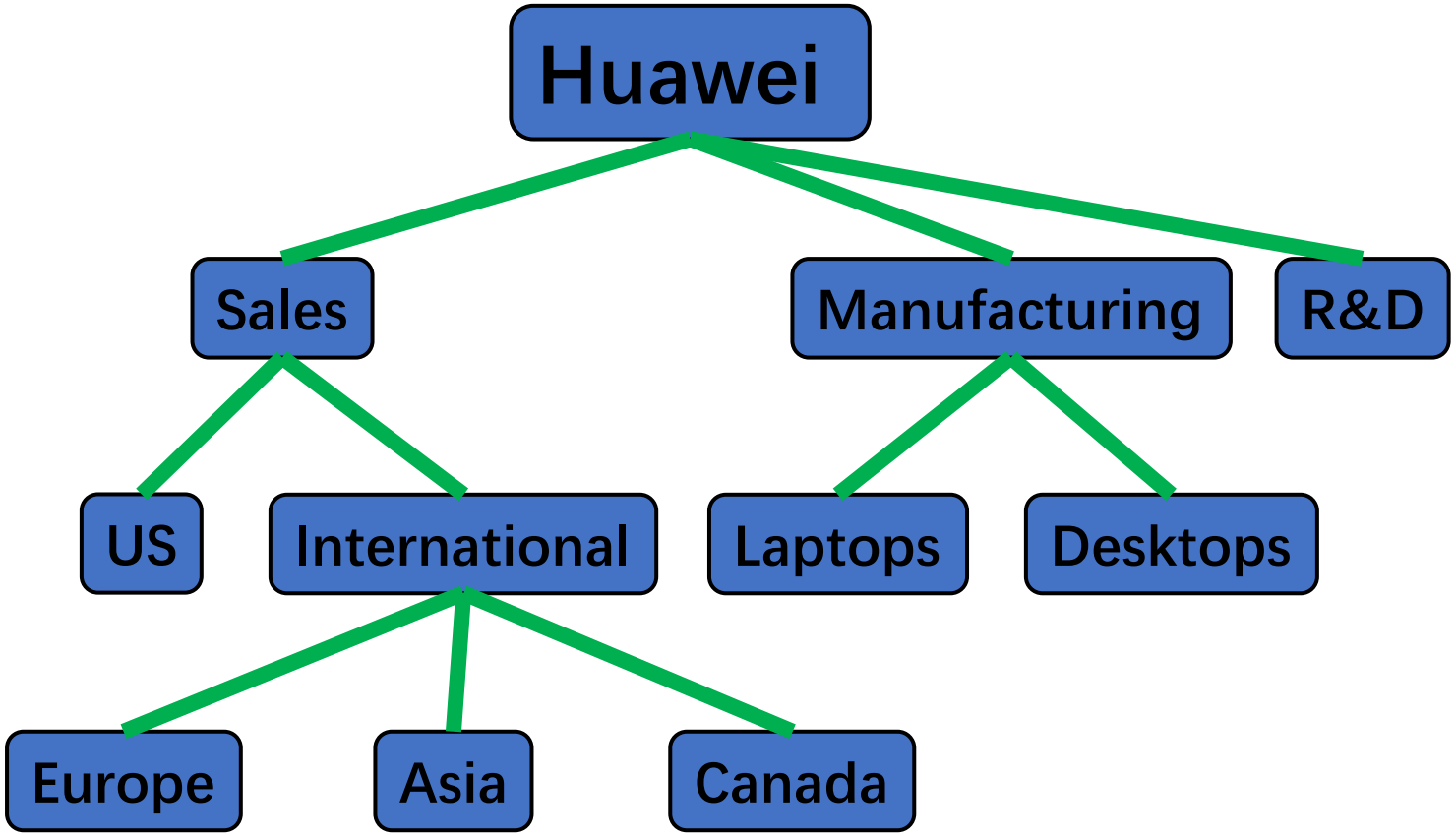
Lecture 10： Data Structure: Trees

By Robert from UOA & Jingong Li from SCNU

華南師範大學
SOUTH CHINA NORMAL UNIVERSITY
UNIVERSITY OF ABERDEEN

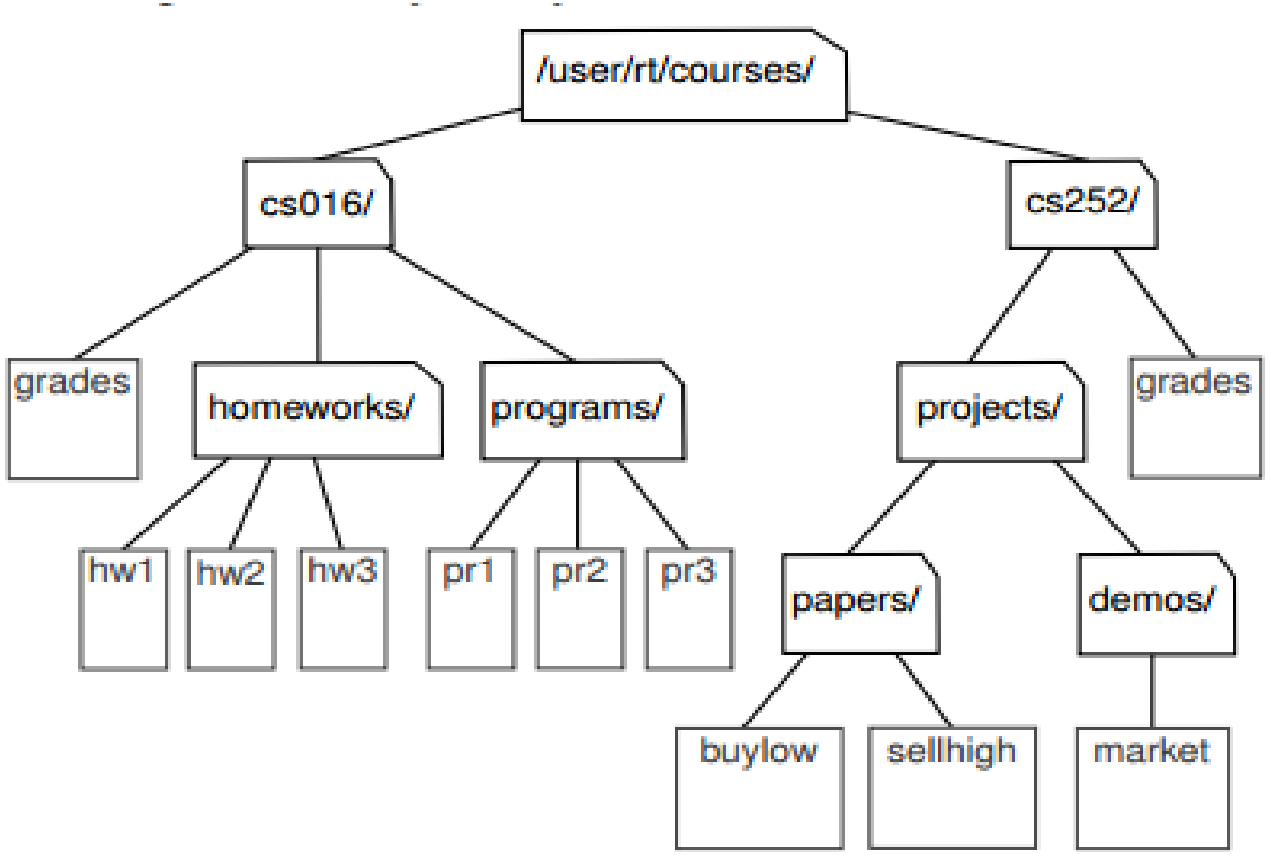
Data Structure: Trees

What is a Tree?



What is a Tree?

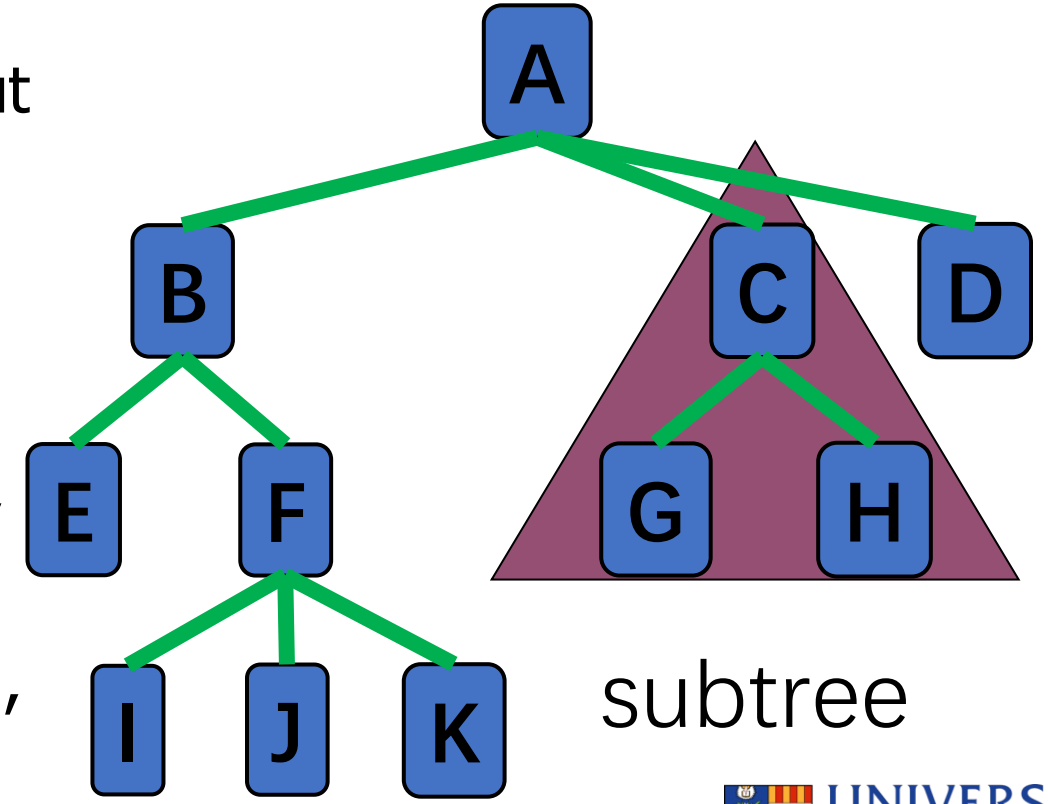
- In computer science, a tree is an abstract model of a hierarchical structure
- A tree consists of nodes with a parent-child relation
- Applications:
  - Organization charts
  - File systems
  - Programming environments



Tree Terminology

- **Root**: node without parent (A)
- **Internal node**: node with at least one child (A, B, C, F)
- **External node** (a.k.a. leaf): node without children (E, I, J, K, G, H, D)
- **Ancestors of a node**: parent, grandparent, grand-grandparent, etc.
- **Depth of a node**: number of ancestors
- **Height of a tree**: maximum depth of any node (3)
- **Descendant of a node**: child, grandchild, grand-grandchild, etc.

- **Subtree**: tree consisting of a node and its descendants



subtree

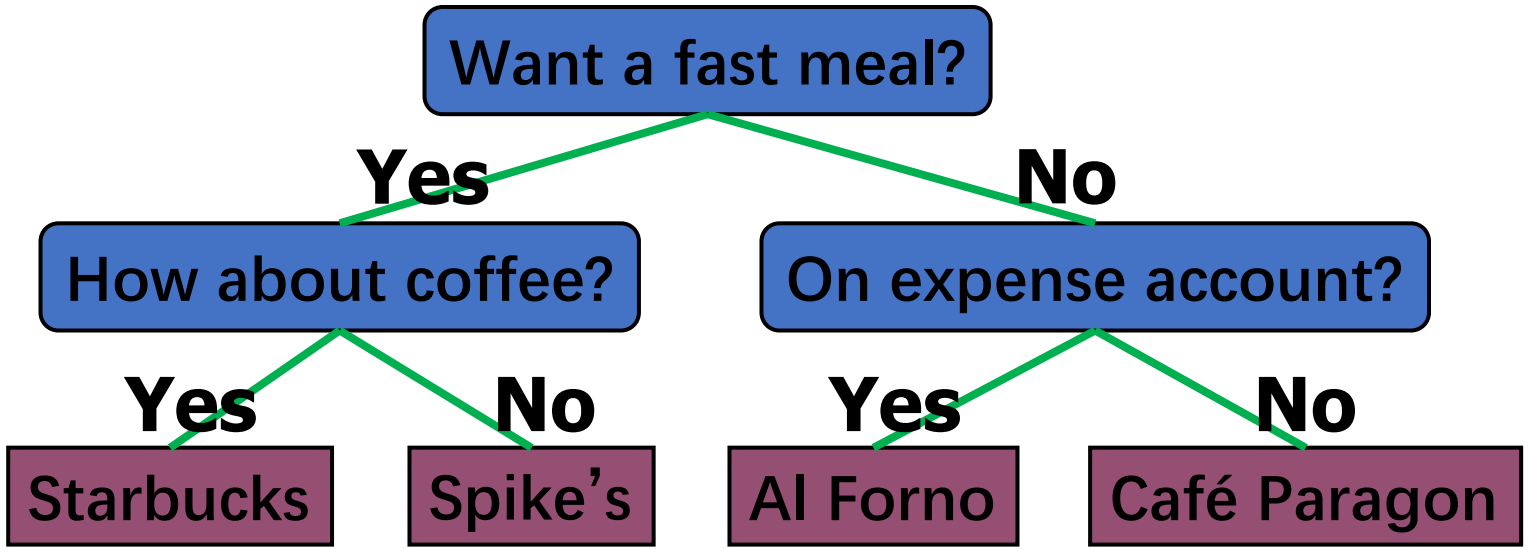
Tree ADT

- We use positions to abstract nodes
- Generic methods:
  - Integer len()
  - Boolean is_empty()
  - Iterator positions()
  - Iterator iter()
- Accessor methods:
  - position root()
  - position parent(p)
  - Iterator children(p)
  - Integer num_children(p)

- Query methods:
  - Boolean is_leaf(p)
  - Boolean is_root(p)
- Update method:
  - element replace (p, o)
- Additional update methods may be defined by data structures implementing the Tree ADT

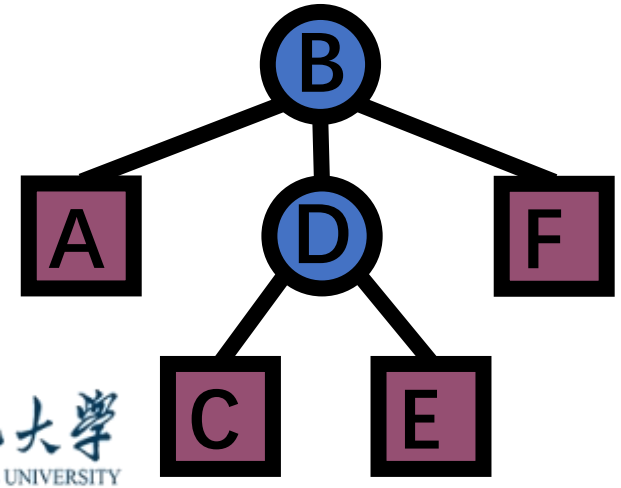
Decision Tree

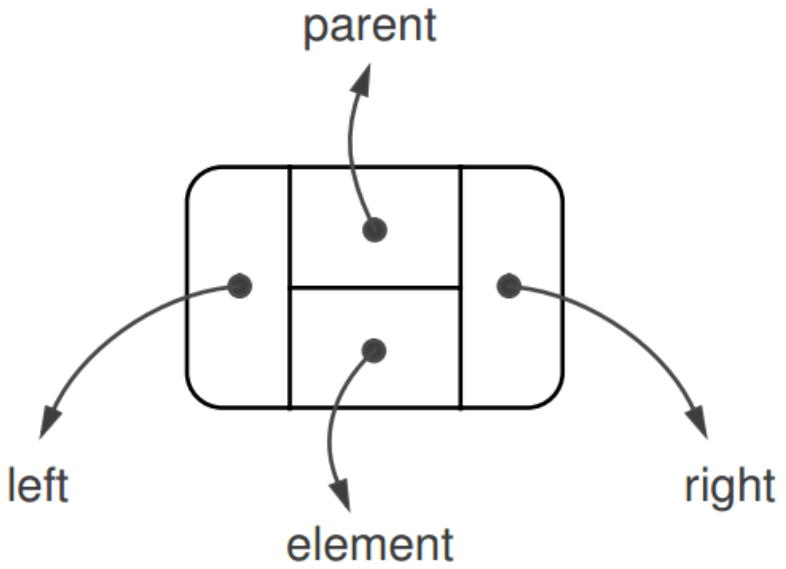
- Binary tree associated with a decision process
  - internal nodes: questions with yes/no answer
  - external nodes: decisions
- Example: dining decision

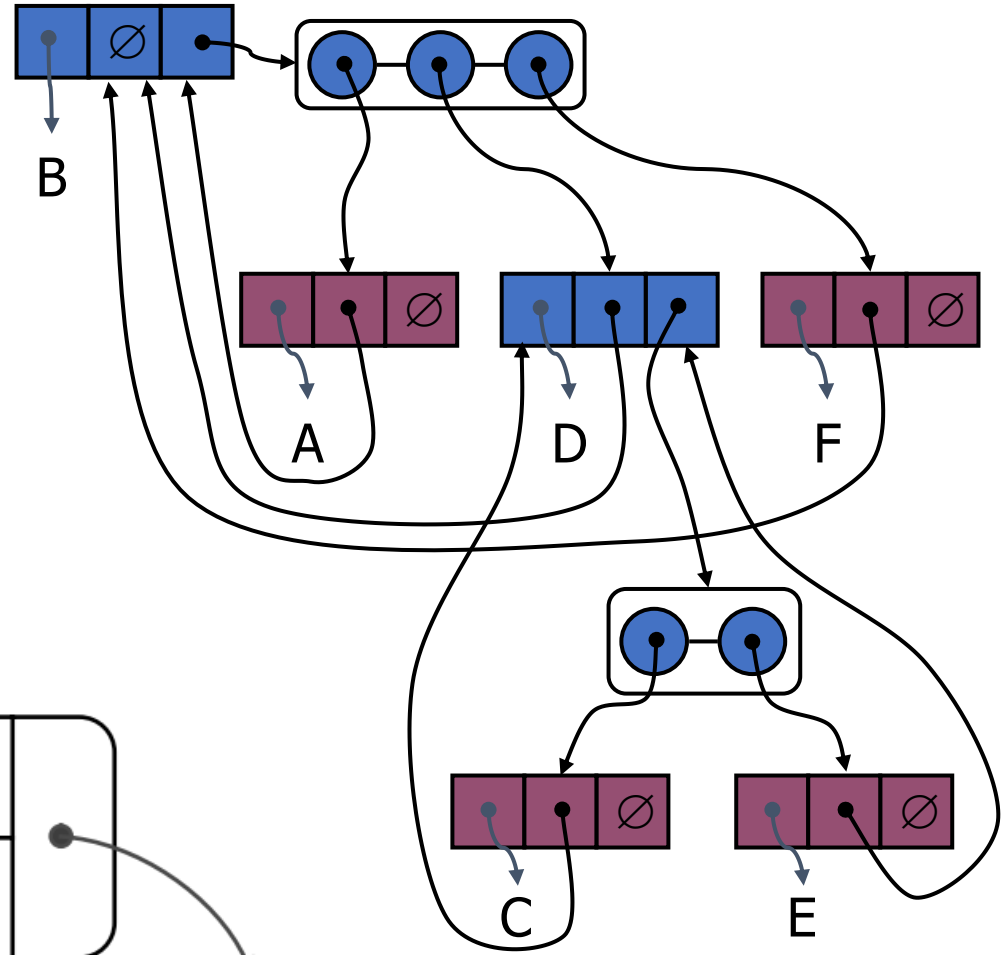


Linked Structure for Trees

- A node is represented by an object storing
  - Element
  - Parent node
  - Sequence of children nodes
- Node objects implement the Position ADT



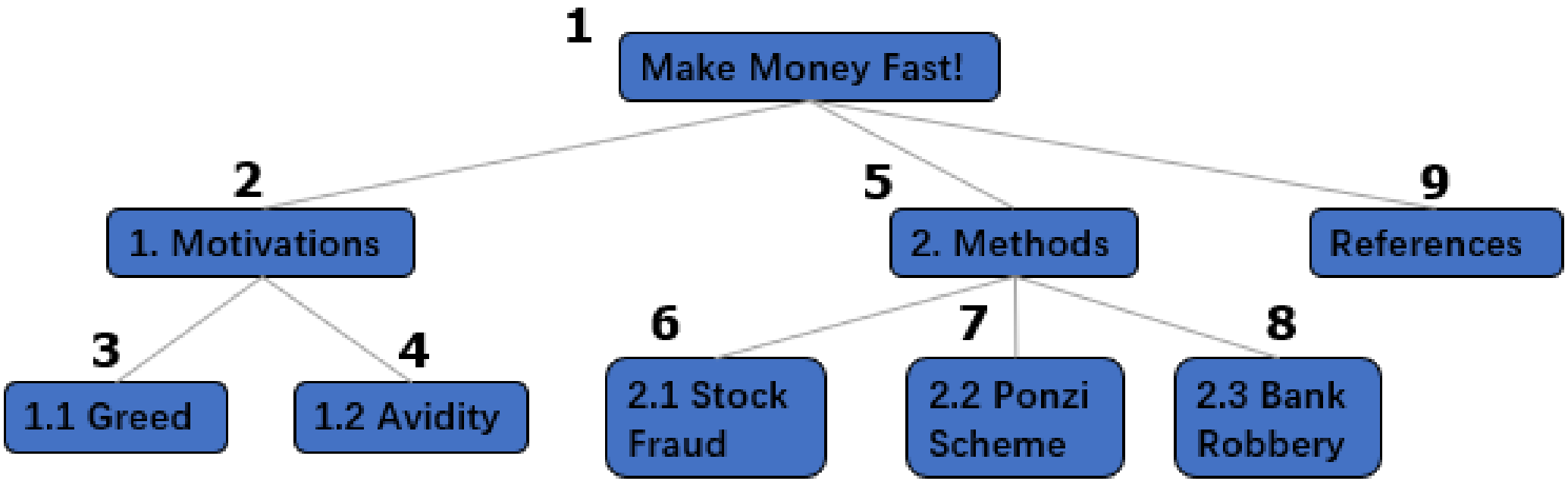




Preorder Traversal

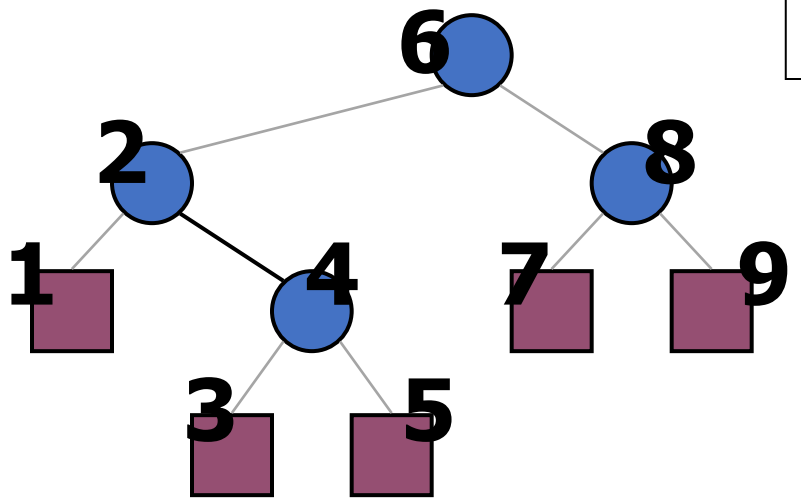
- A traversal visits the nodes of a tree in a systematic manner
- In a preorder traversal, a node is visited before its descendants
- Application: print a structured document

Algorithm ***preOrder***(***v***)
    visit(***v***)
    for each child ***w*** of ***v***
        preorder (***w***)



Inorder Traversal

- In an inorder traversal a node is visited after its left subtree and before its right subtree
- Application: draw a binary tree
  - x(v) = inorder rank of v
  - y(v) = depth of v

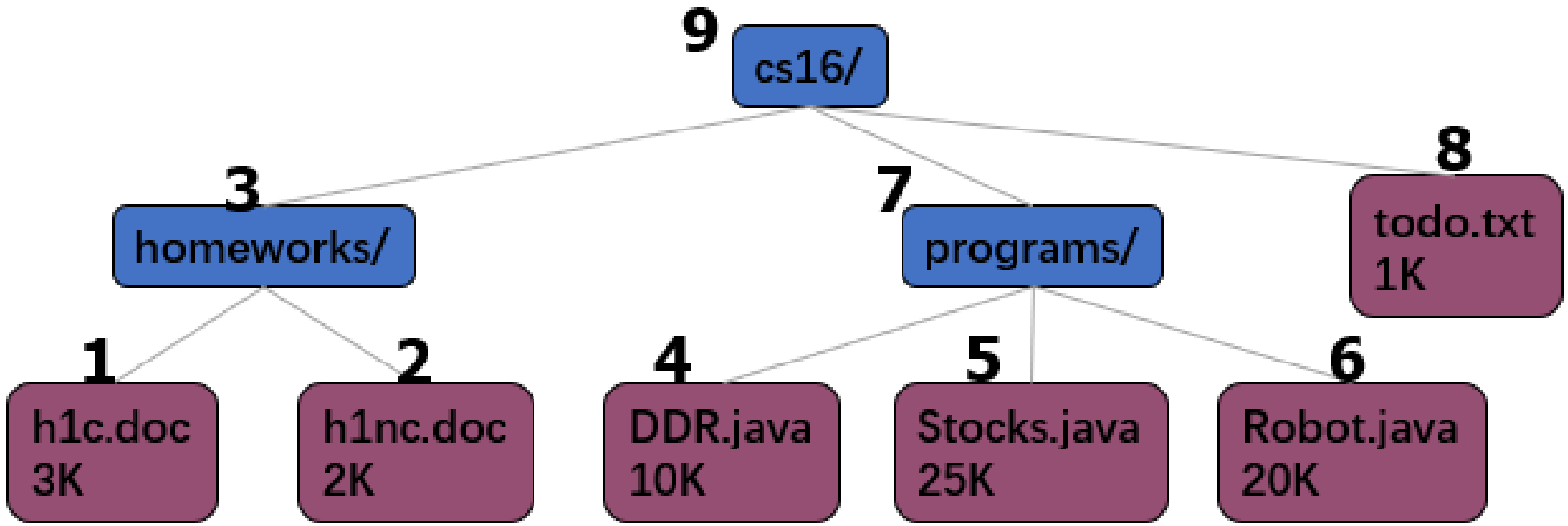


Algorithm ***inOrder***(***v***)
  if ***v*** **has a left child**
    inOrder (***left*** (***v***))
  visit(***v***)
  if ***v*** **has a right child**
    inOrder (***right*** (***v***))

Postorder Traversal

- In a postorder traversal, a node is visited after its descendants
- Application: compute space used by files in a directory and its subdirectories

Algorithm ***postOrder***(***v***)
  for each child ***w*** of ***v***
    postOrder (***w***)
  visit(***v***)



Pre/In/Post-order Traversal

Preorder先序遍历：根——左子树——右子树

Inorder中序遍历：左子树——根——右子树

Postorder后序遍历：左子树——右子树——根

Thank You