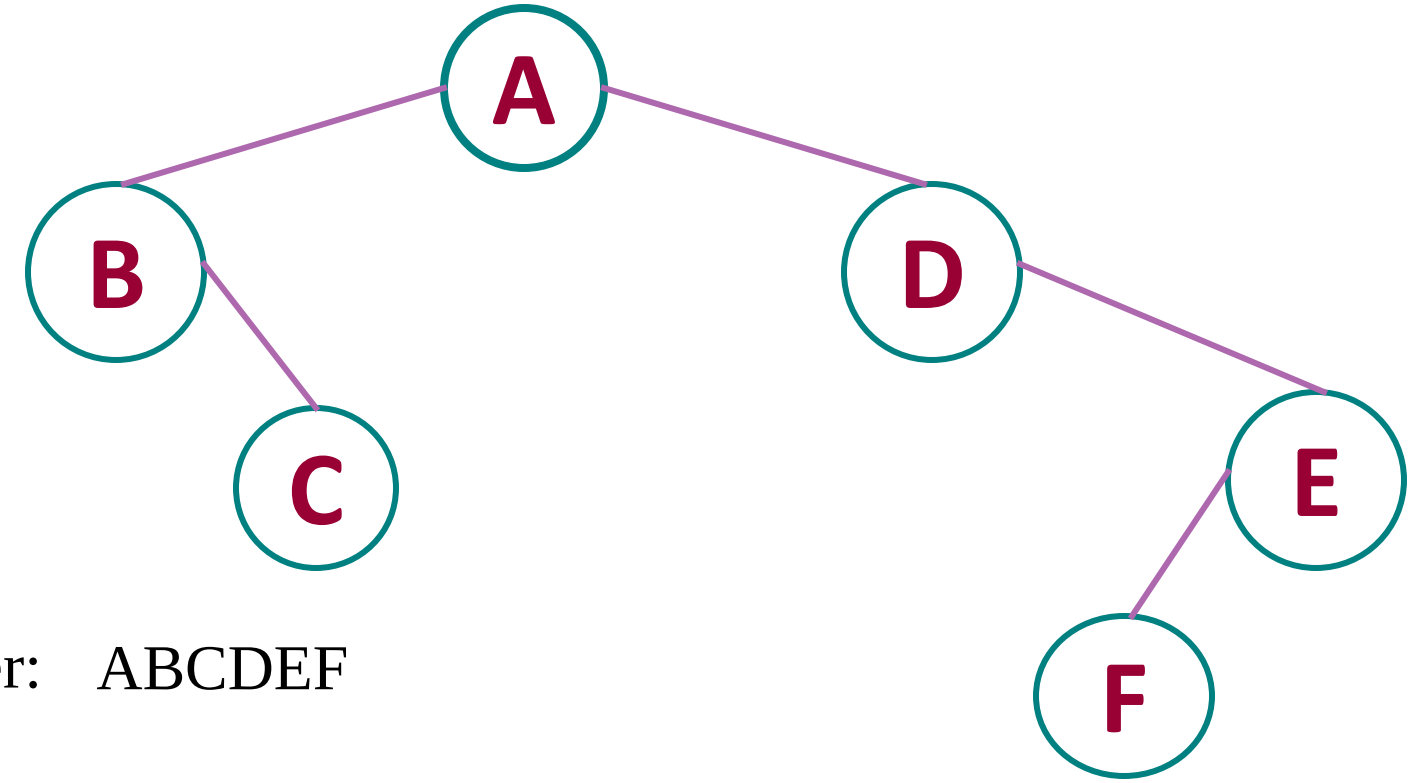
Binary Trees

- Traversal of Binary Trees
  - At a given node there are three tasks to do in some order: Visit（访问） the node itself (*V*); traverse its left subtree（左子树） (*L*); traverse its right subtree（右子树） (*R*).

    V L R *L V R* *L R V*

    preorder（前序） *inorder*（中序） *postorder*（后序）
  - With *preorder traversal* we first visit a node, then traverse its left subtree, and then traverse its right subtree.
  - With *inorder traversal* we first traverse the left subtree, then visit the node, and then traverse its right subtree.
  - With *postorder traversal* we first traverse the left subtree, then traverse the right subtree, and finally visit the node.

举例：

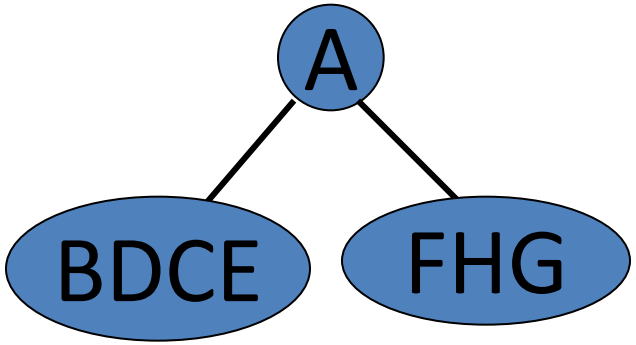


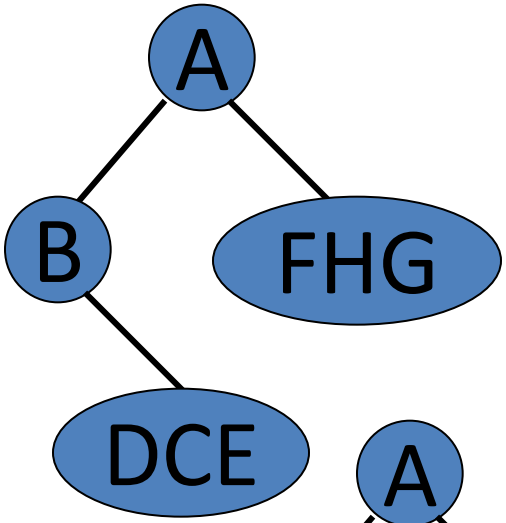
preorder: ABCDEF

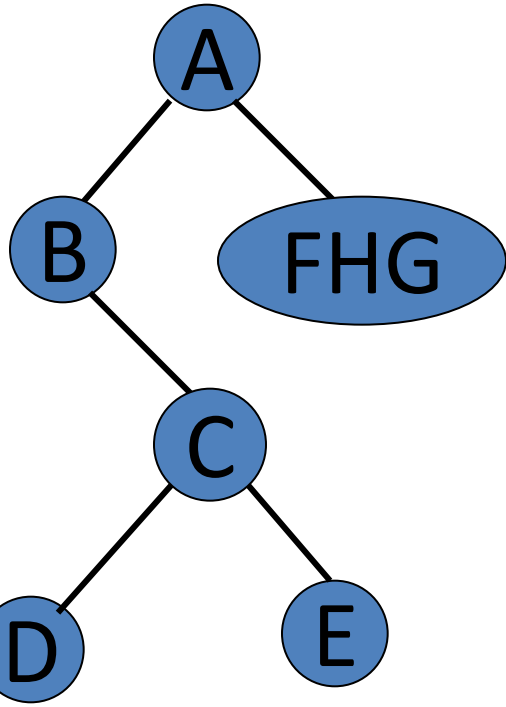
inorder: BCADFE

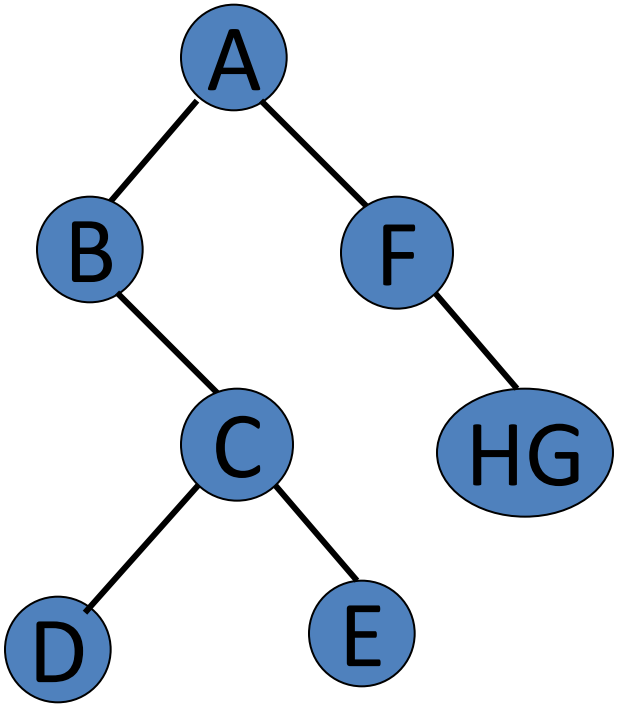
postorder: CBFEDA

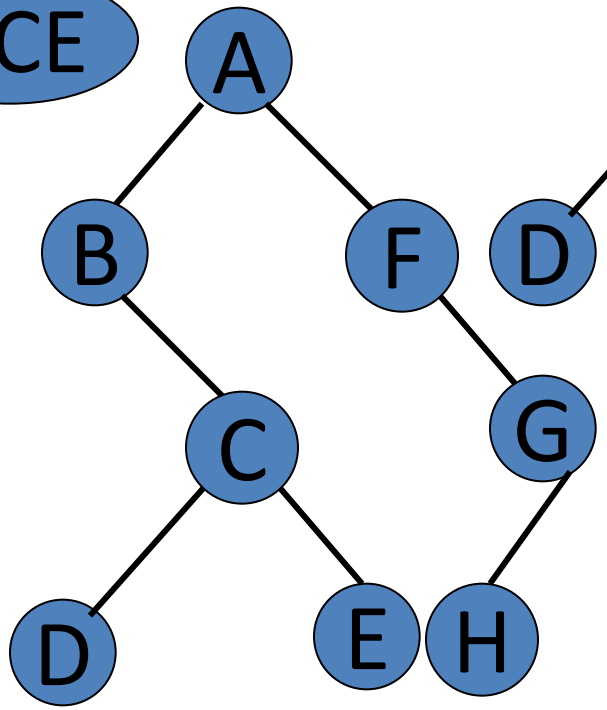
已知一棵二叉树的先根序列和中根序列分别为ABCDEFGH和BDCEAFHG 。画出此二叉树。





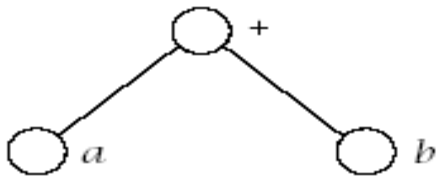




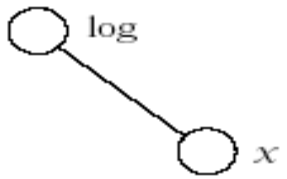


Binary Trees

- Expression Trees

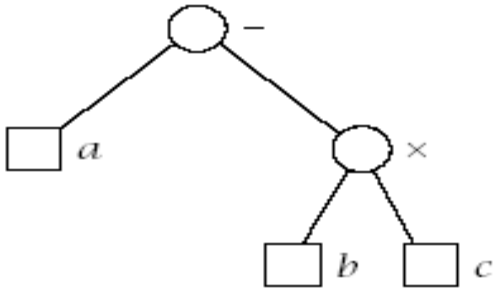


$a + b$

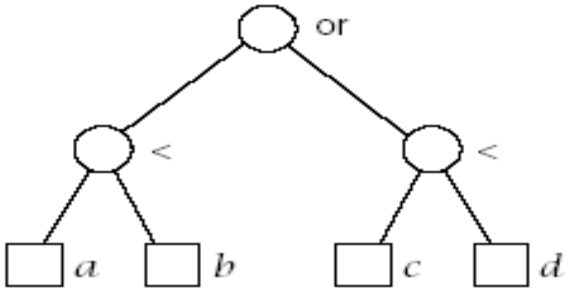


$\log x$

$n!$



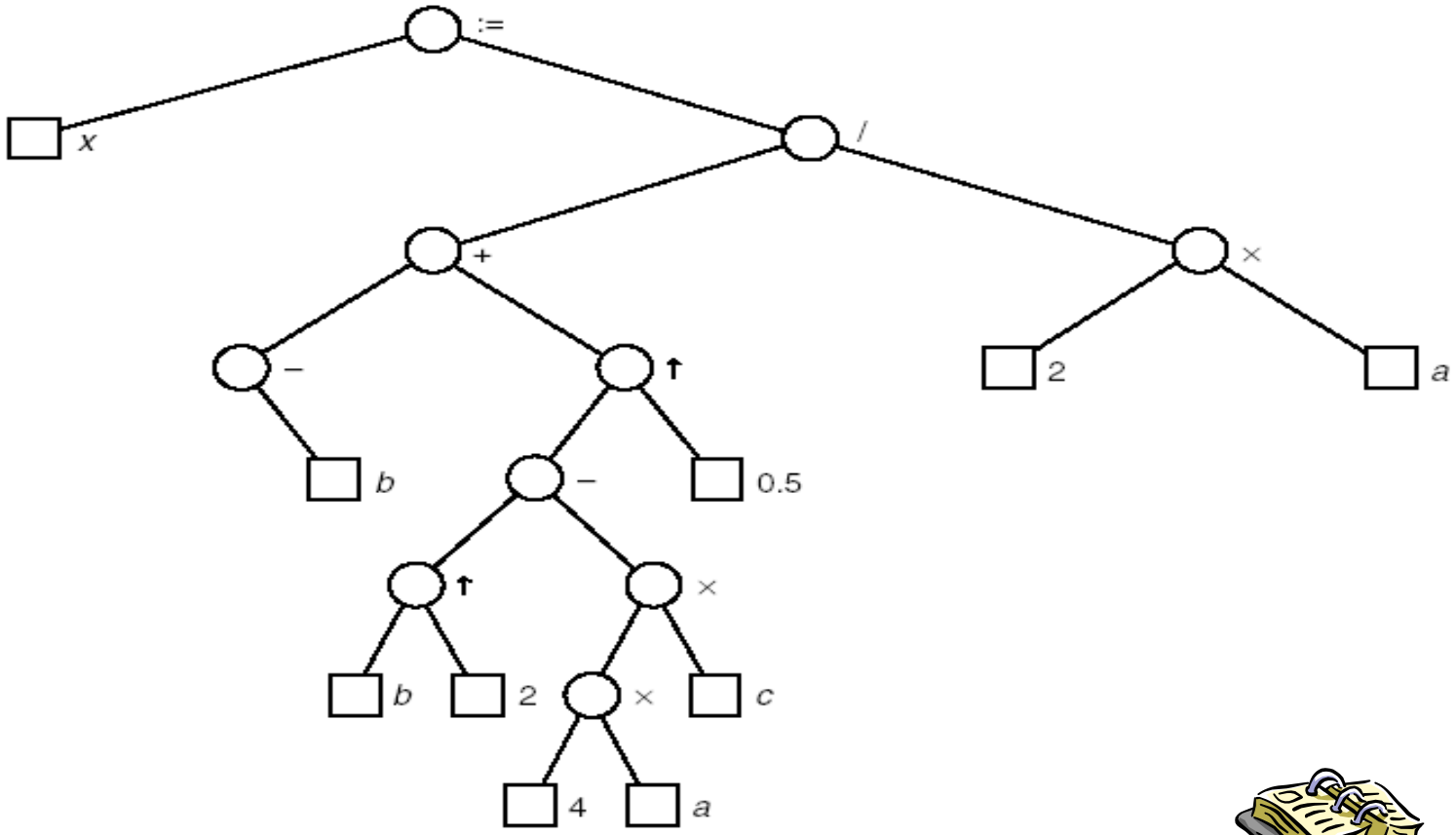
$a - (b \times c)$



$(a < b)$ or $(c < d)$

Binary Trees

- Expression Trees



$x := (-b + (b \uparrow 2 - 4 \times a \times c) \uparrow 0.5)/(2 \times a)$

对树的遍历操作	表达式的形式
前序序列	前缀表达式
中序序列	中缀表达式
后序序列	后缀表达式

对二分查找的比较树的中序遍历得到有序序列

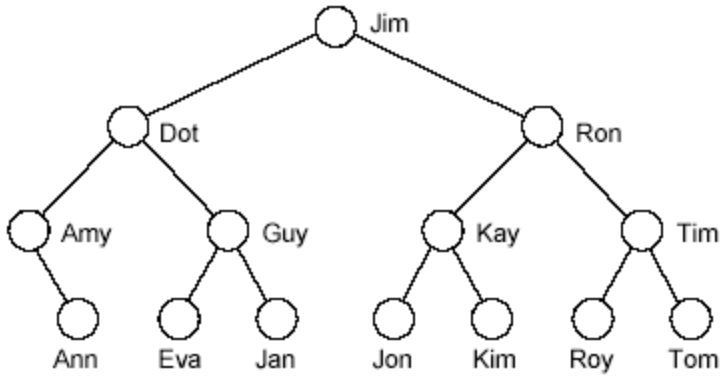


Figure 10.1. Comparison tree for binary search