

二叉树

后序遍历：观察

05 - G1

看了，因向仙姑道：“敢烦仙姑引我到那各司中游玩游玩，不知可使得？”

仙姑道：“此各司中皆贮的是普天之下所有的女子过去未来的簿册，尔凡眼尘躯，未便先知的。”

邓俊辉

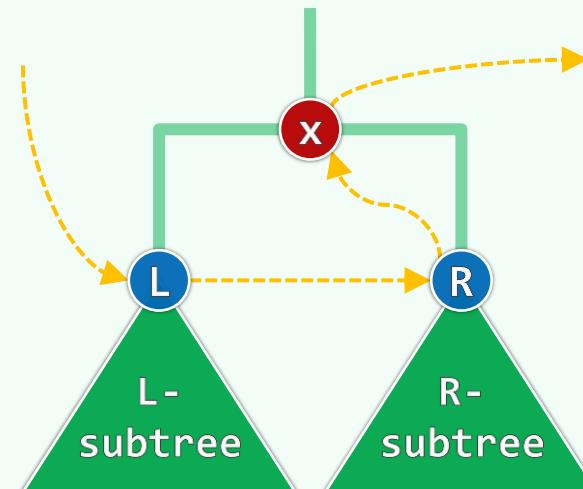
deng@tsinghua.edu.cn

# 递归实现

❖ 应用：BinNode::size() + BinTree::updateHeight()

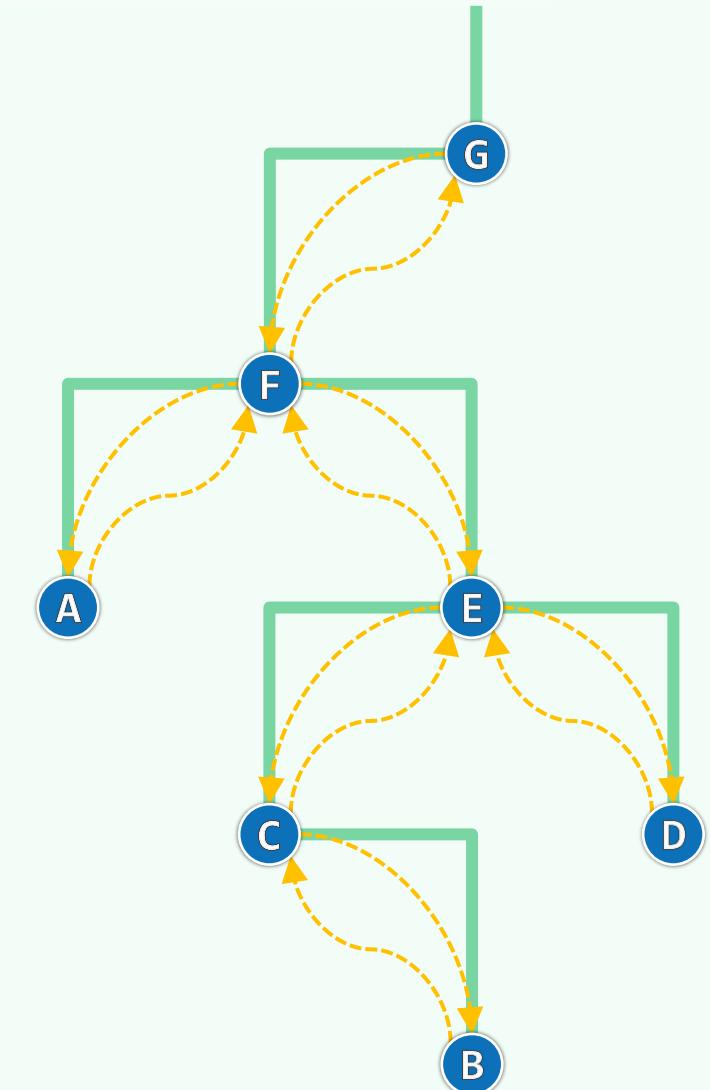
❖ template <typename T, typename VST>

```
void traverse( BinNodePosi(T) x, VST & visit ) {  
    if ( ! x ) return;  
  
    traverse( x->lc, visit );  
  
    traverse( x->rc, visit );  
  
    visit( x->data );  
}
```



$$\diamond T(n) = \mathcal{O}(1) + T(a) + T(n - a - 1) = \mathcal{O}(n)$$

❖ 挑战：不依赖递归机制，能否实现后序遍历？如何实现？效率如何？



# 思路

◆ 对左、右子树的递归遍历，都不是尾递归

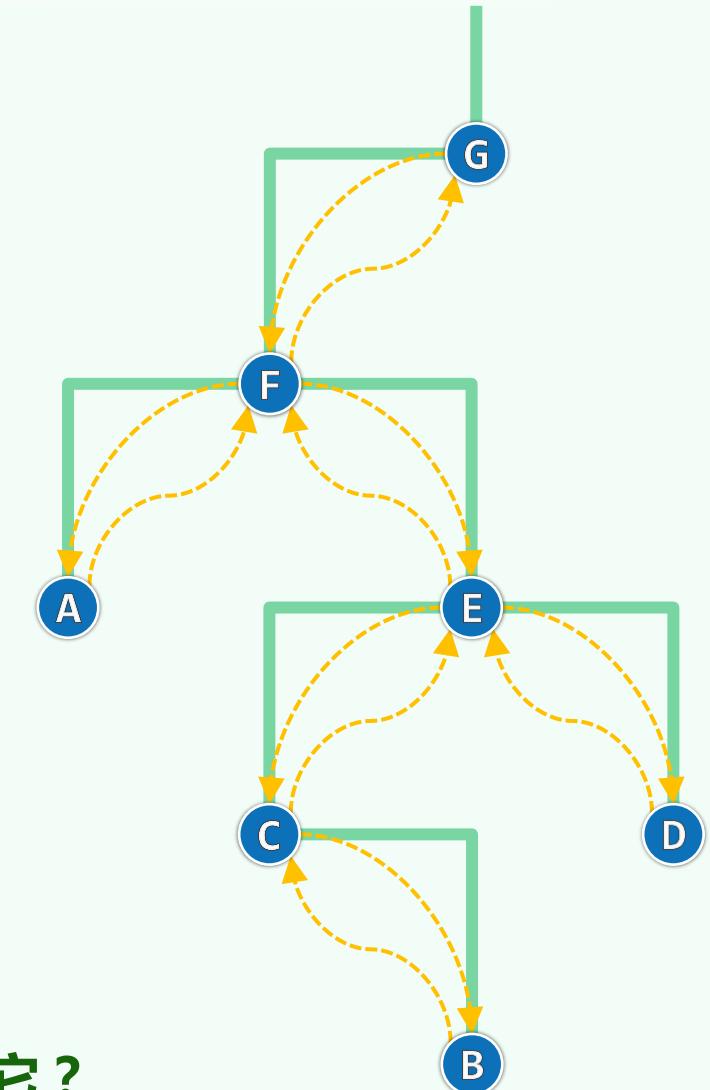
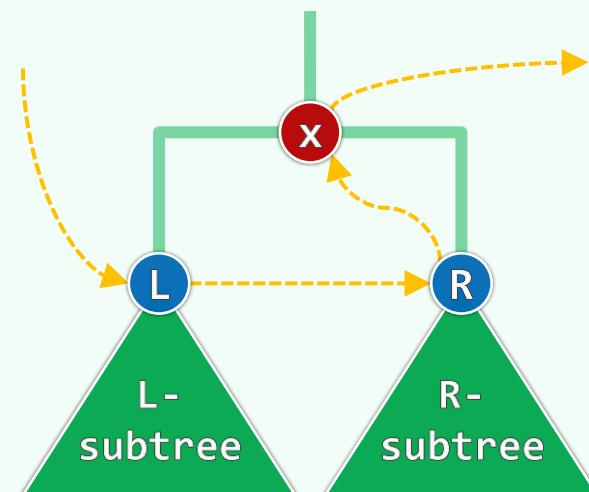
◆ 解决方法：找到第一个被访问的节点，用栈保存其祖先及右兄弟

◆ 这样，原问题就被分解为

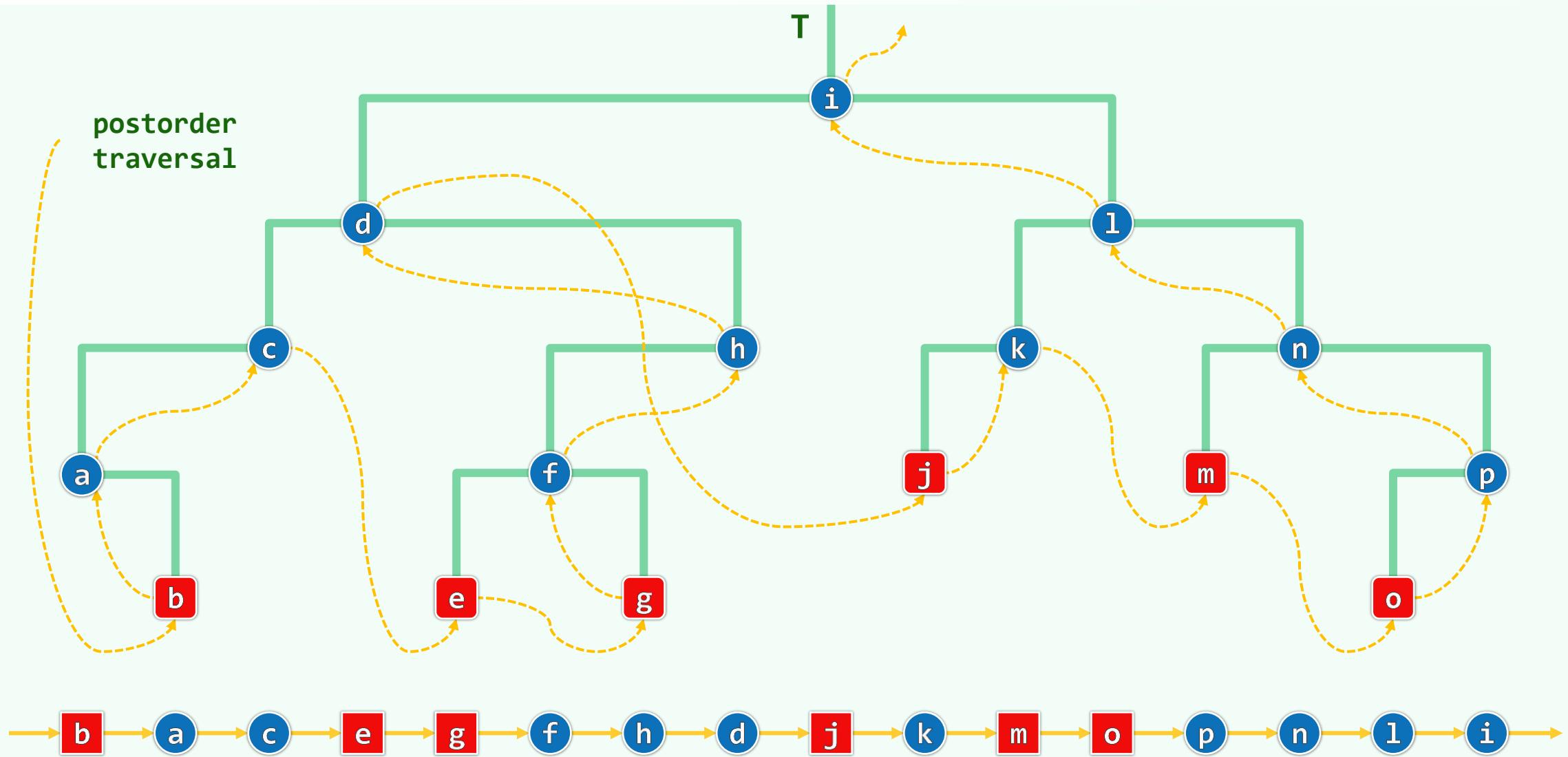
- 依次对若干棵右子树的遍历问题

◆ 于是，首先要解决的问题仍是

- 后序遍历任一二叉树T时，第一个被访问的是哪个节点？如何找到它？



# 观察



# 藤缠树

❖ 从根出发下行

尽可能沿左分支

实不得已，才沿右分支

❖ 最后一个节点

必是叶子，而且

是按中序遍历次序最靠左者

也是递归版中visit()首次执行处

❖ 这匹叶子，将首先接受访问...

