

5. 二叉树

后序遍历

观察

当下随了仙姑进入二层门内，至两边配殿，皆有匾额对联，一时看不尽许多，惟见有几处写的是：“痴情司”、“结怨司”、“朝啼司”、“夜怨司”、“春感司”、“秋悲司”。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

❖ template <typename T, typename VST>

```
void traverse( BinNodePosi(T) x, VST & visit ) {
```

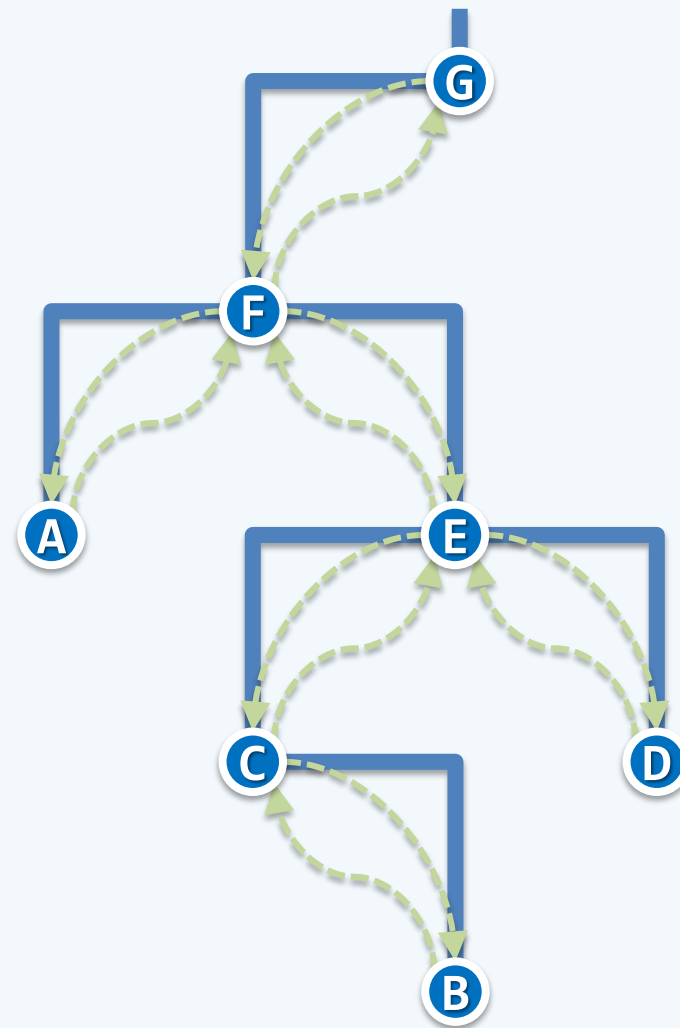
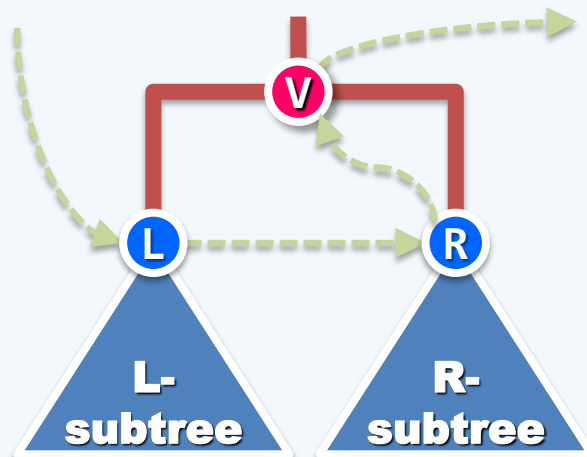
```
    if ( ! x ) return;
```

```
    traverse( x->lc, visit );
```

```
    traverse( x->rc, visit );
```

```
    visit( [x]->data );
```

```
} //T(n) = T(a) + T(n - a - 1) + O(1) = O(n)
```



❖ 应用 : BinNode::size()

BinTree::updateHeight()

❖ 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 ?

难点

❖ 难度在于

- 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
- 将其祖先及其右兄弟（如果存在）用栈保存

❖ 这样，原问题就被分解为

- **依次**对若干棵右子树的遍历问题

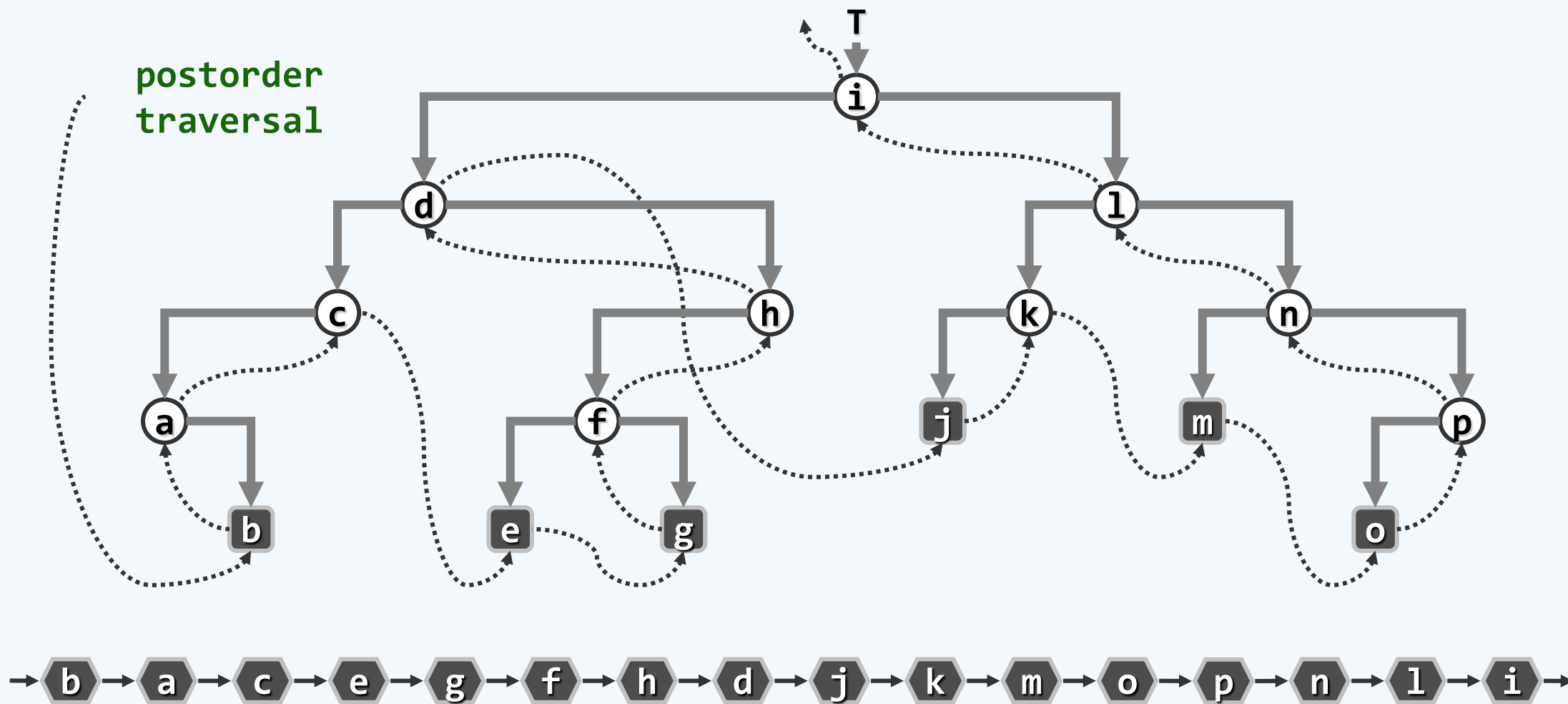
//同样地，这里应依什么“次”？

❖ 于是，首先要解决的问题仍是

- 后序遍历任一二叉树T时，**首先被访问**的是哪个节点？如何找到它？

观察

postorder
traversal



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

是从左侧可见的最高叶子

❖ 这匹叶子，将首先接受访问

