05-F1

二叉树

中序遍历：观察

山中只见藤缠树，世上哪见树缠藤

青藤若是不缠树，枉过一春又一春

邓俊辉

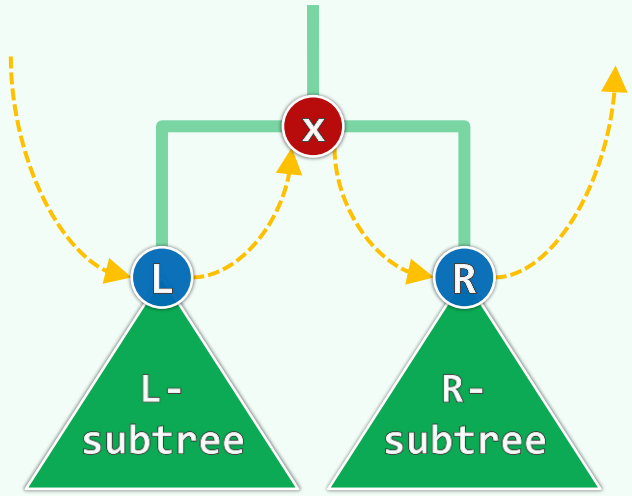
deng@tsinghua.edu.cn

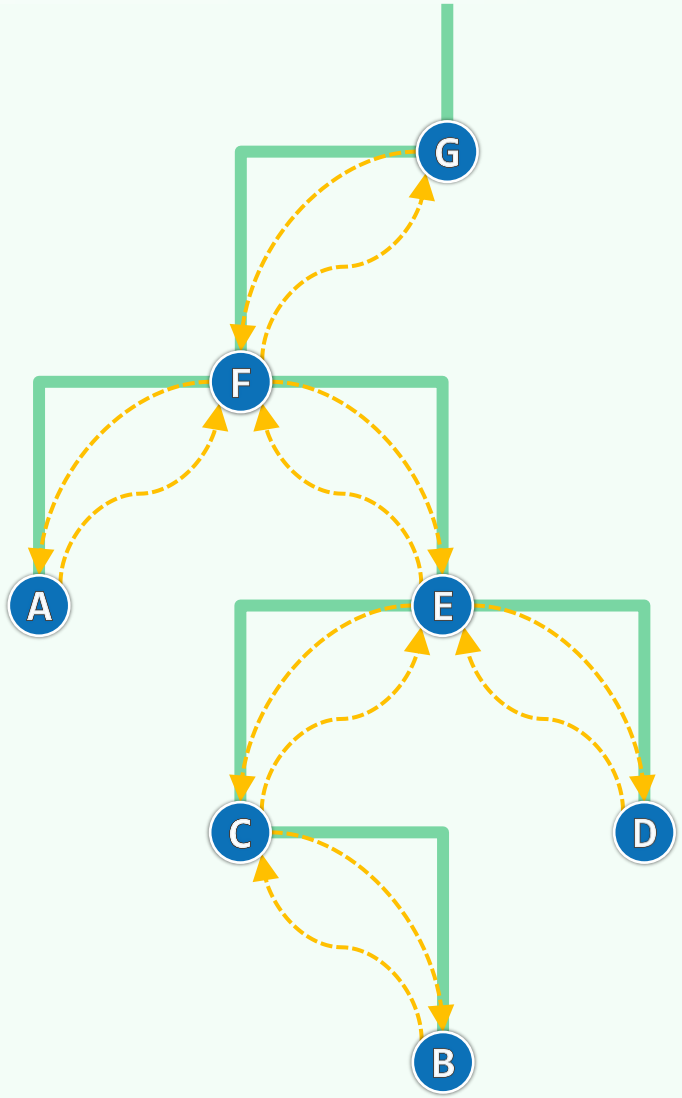
递归实现

❖ 应用：中序输出文件树结构：printBinTree()

❖ template <typename T, typename VST>

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





❖ T(n) = O(1) + T(a) + T(n – a - 1) = O(n)

❖ 挑战： 不依赖递归机制，能否实现中序遍历？如何实现？效率如何？

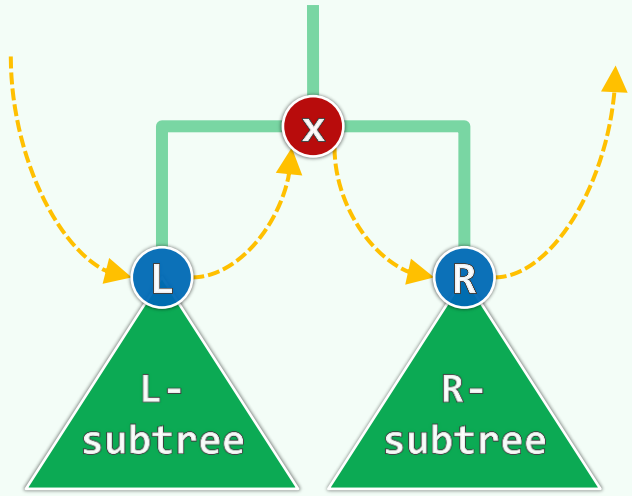
思路

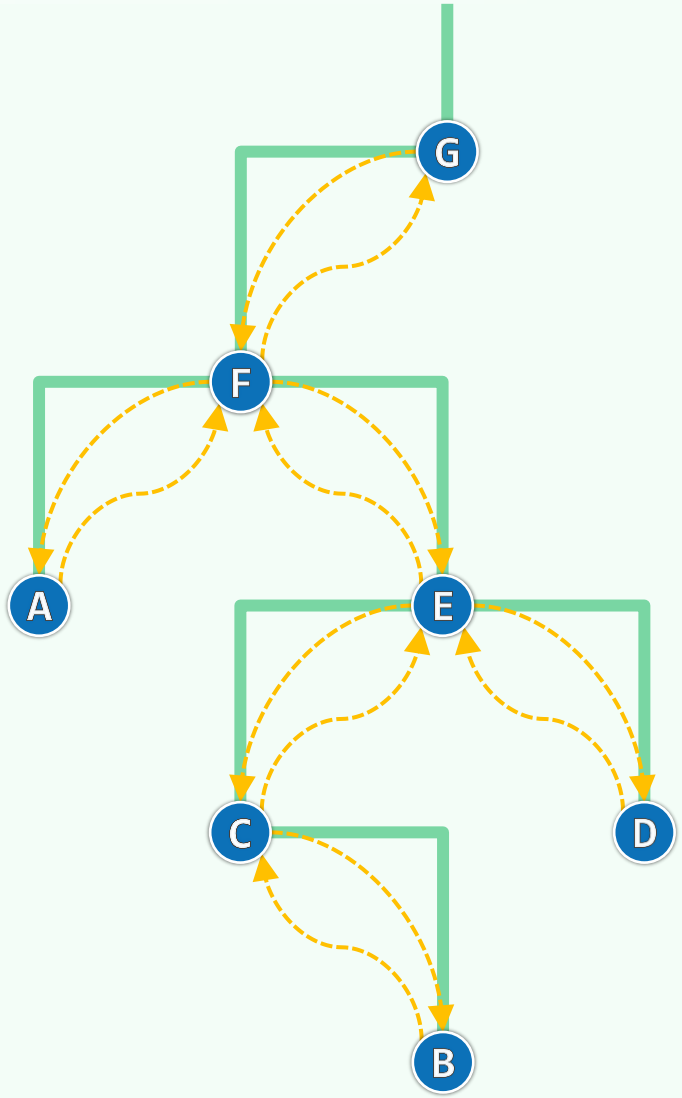
- 难度在于：尽管右子树的递归遍历是尾递归，但左子树却严格地不是
- 解决方法：找到第一个被访问的节点，用栈保存其祖先
- 这样，原问题就被分解为

  依次对若干棵右子树的遍历问题？
- 于是，首先要解决的问题就是：

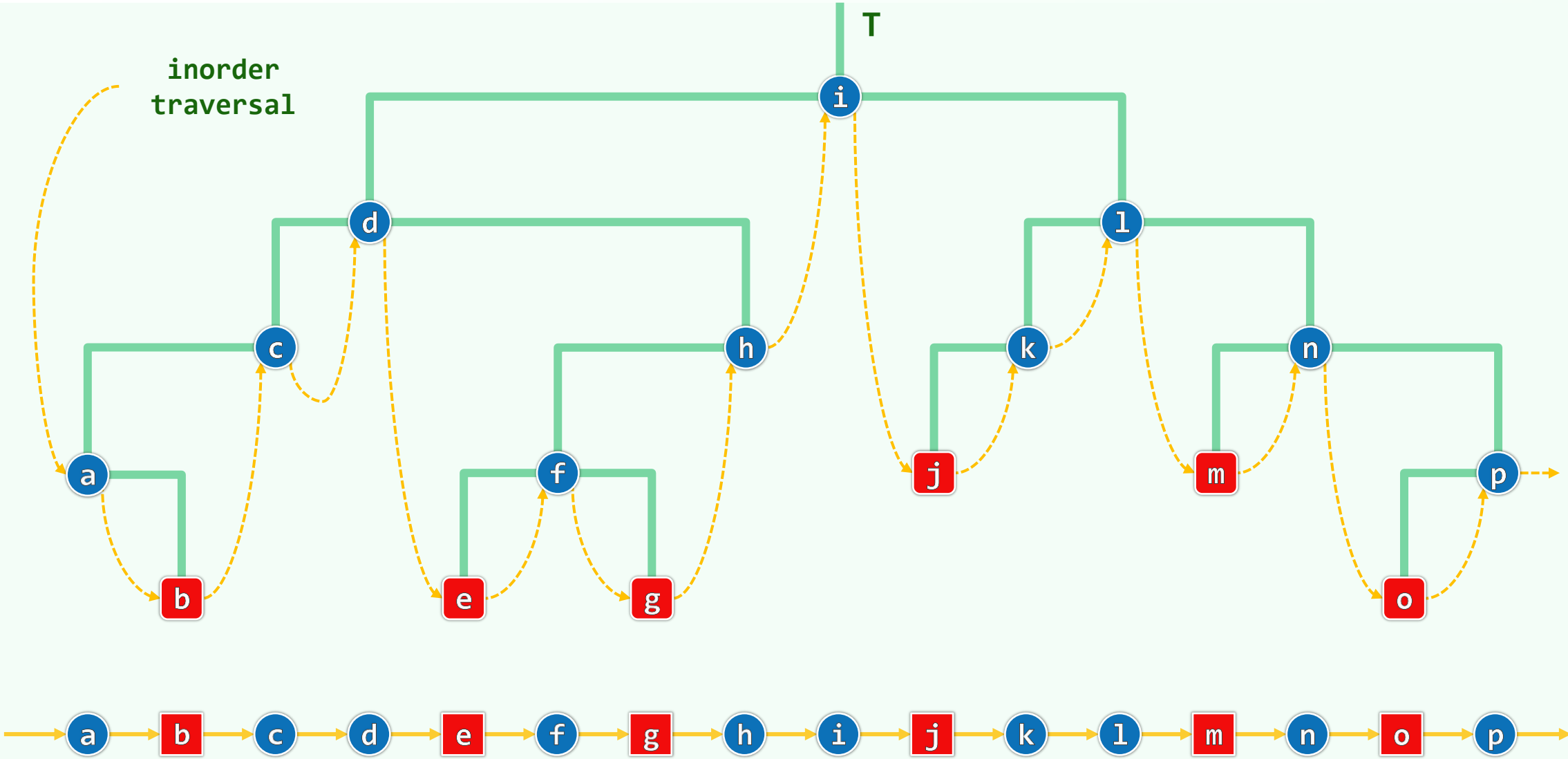
  中序遍历任一二叉树T时

  第一个被访问的是哪个节点？如何找到它？





观察



藤缠树

- ❖ 沿着左侧分支（藤），整个遍历过程可分解为自底而上的d+1步迭代
  - 访问藤上节点
  - 再遍历其右子树
- ❖ 各右子树的遍历彼此独立

  自成一个子任务