05-E1

二叉树

先序遍历：观察

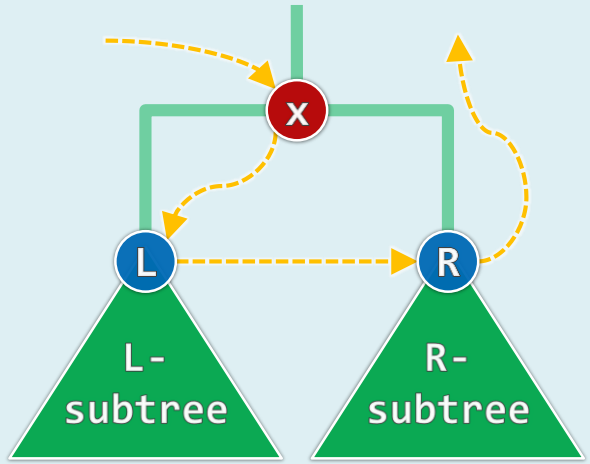
一桩事情的真相与奥妙，通常并不藏在最深的地方，有时就在表面。只不过，一般人视若无睹。要想成为一个好的算命先生，首先就必须学会观察...

邓俊辉
deng@tsinghua.edu.cn

遍历：按照某种次序访问树中各节点，每个节点被访问恰好一次

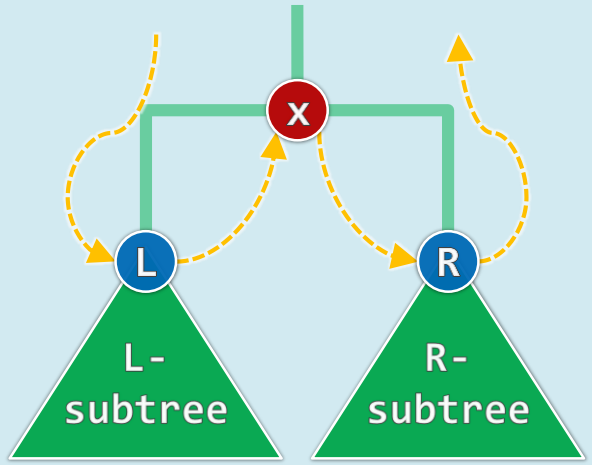
❖ T = L ∪ x ∪ R

先序
preorder



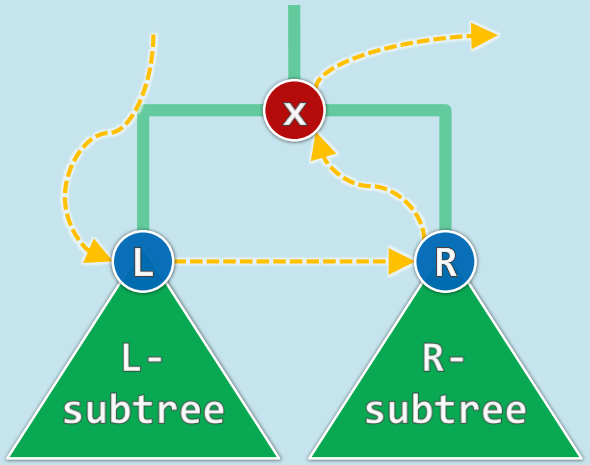
x | L | R

中序
inorder



L | x | R

后序
postorder



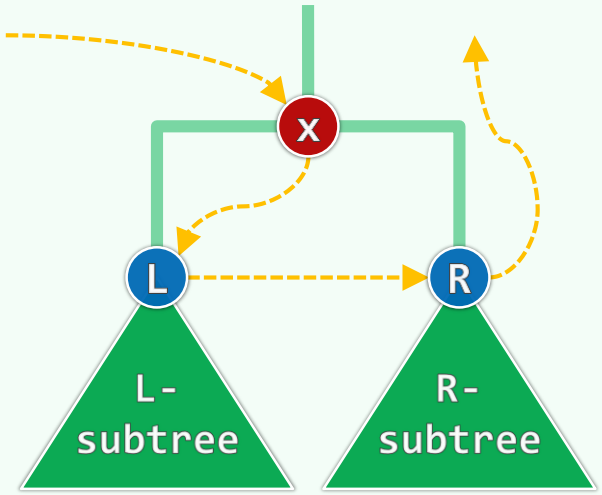
L | R | x

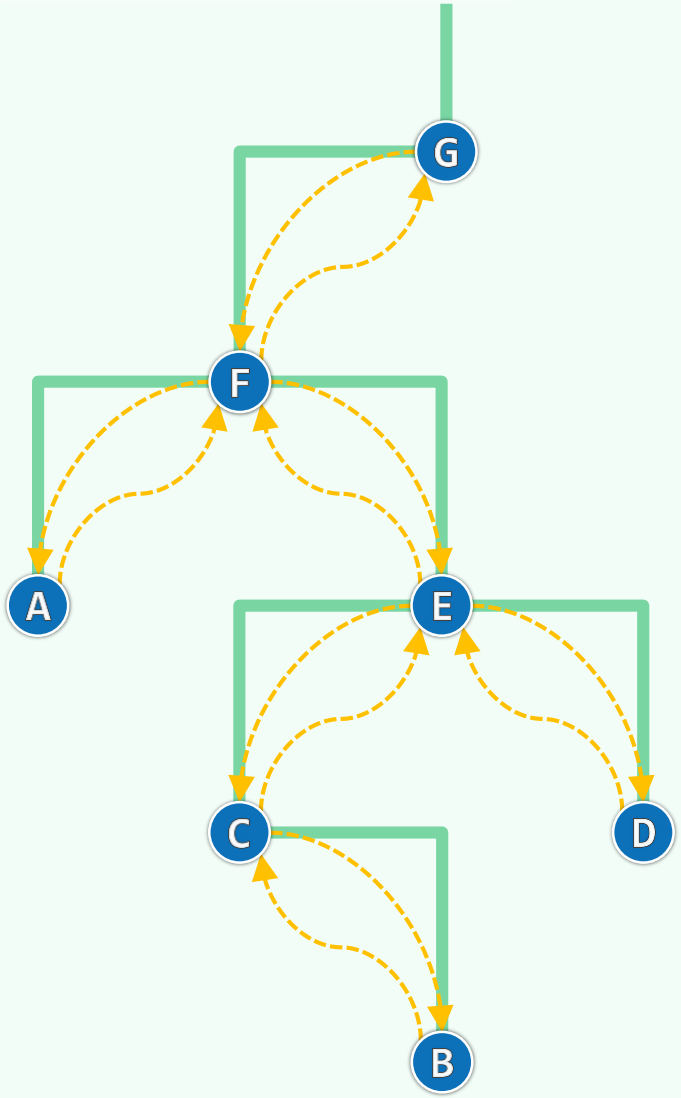
❖ 遍历：结果 ~ 过程 ~ 次序 ~ 策略

递归实现

❖ 应用：先序输出文件树结构：`c:\> tree.com c:\windows`

❖
```
template <typename T, typename VST>
void traverse( BinNodePosi<T> x, VST & visit ) {
    if ( ! x ) return;
    visit( x->data );
    traverse( x->lc, visit );
    traverse( x->rc, visit );
} //O(n)
```

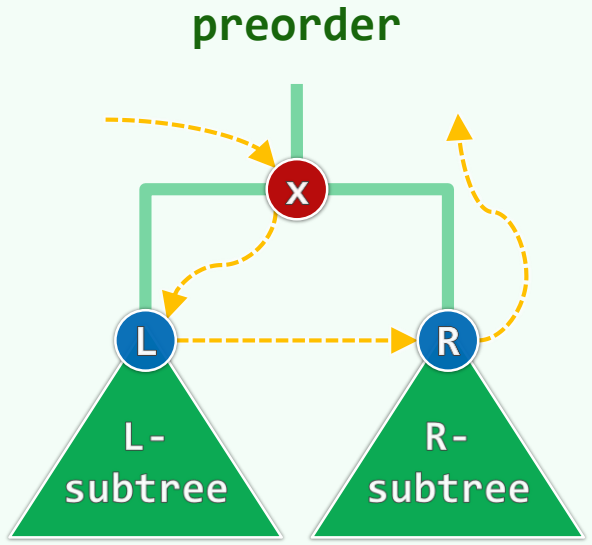




❖ 制约：使用默认的Call Stack，允许的递归深度有限

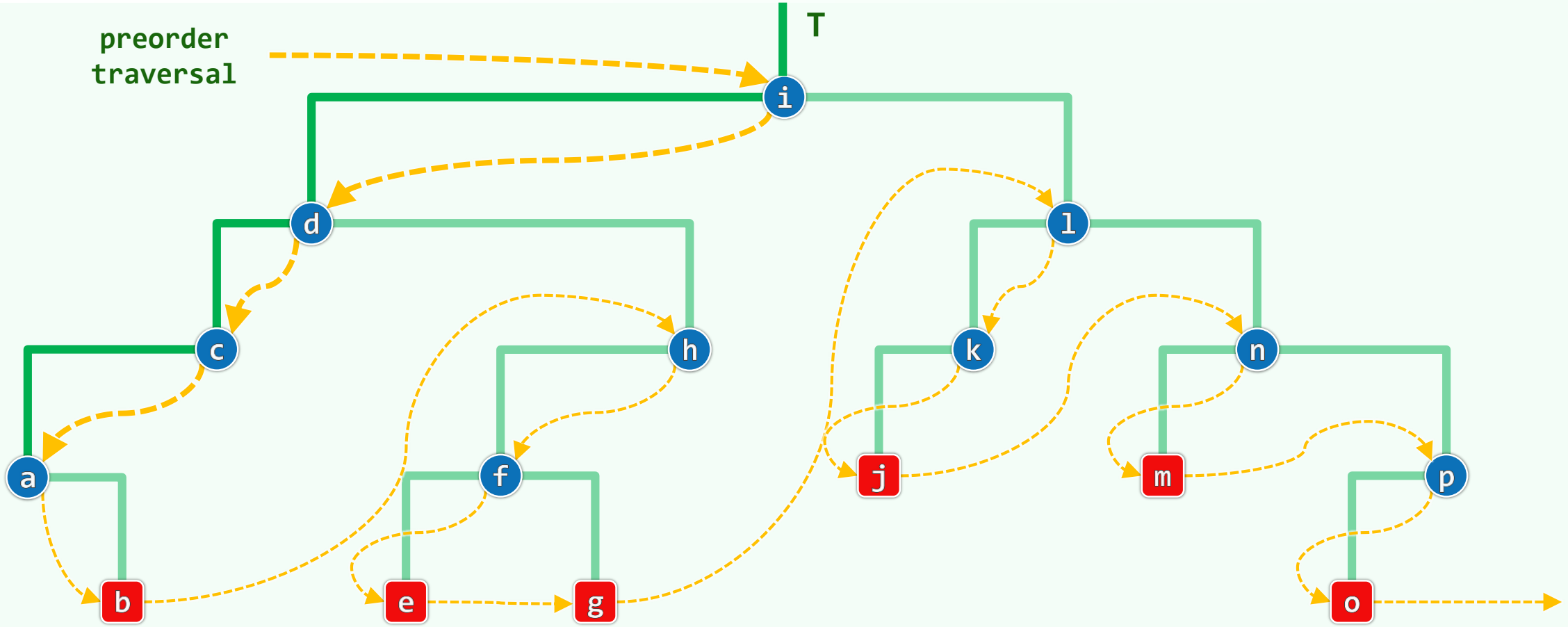
❖ 挑战：不依赖递归机制，能否实现先序遍历？如何实现？效率如何？

先序实例：统计规模




```
template <typename T>

Rank BinNode<T>::size() { //后代总数

   Rank s = 1; //计入本身

   if (lc) s += lc->size(); //递归计入左子树规模

   if (rc) s += rc->size(); //递归计入右子树规模

   return s;

} //懒惰策略，O( n = |size| )
```

观察


preorder
traversal
T
i
d
l
c
h
k
n
a
f
j
m
p
b
e
g
o


i
d
c
a
b
h
f
e
g
l
k
j
n
m
p
o

藤 = 起始于根的左侧通路 = 长子通路

- 沿着藤，整个遍历过程可分解为两个阶段：
  - 自上而下访问藤上节点，再
  - 自下而上遍历各右子树
- 各右子树的遍历彼此独立

  自成一个子任务