05-I

二叉树

重构

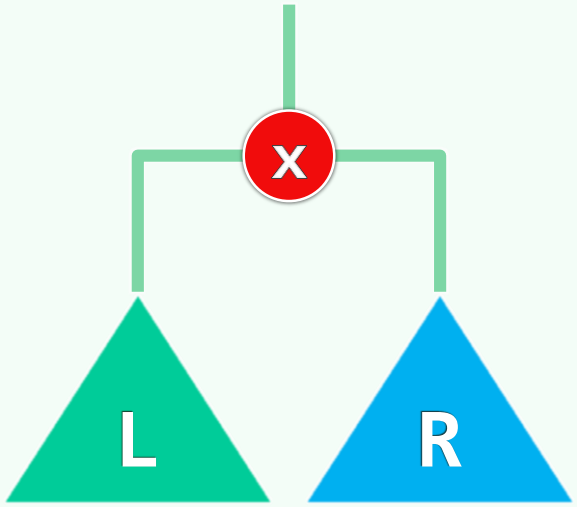
No matter where they take us,

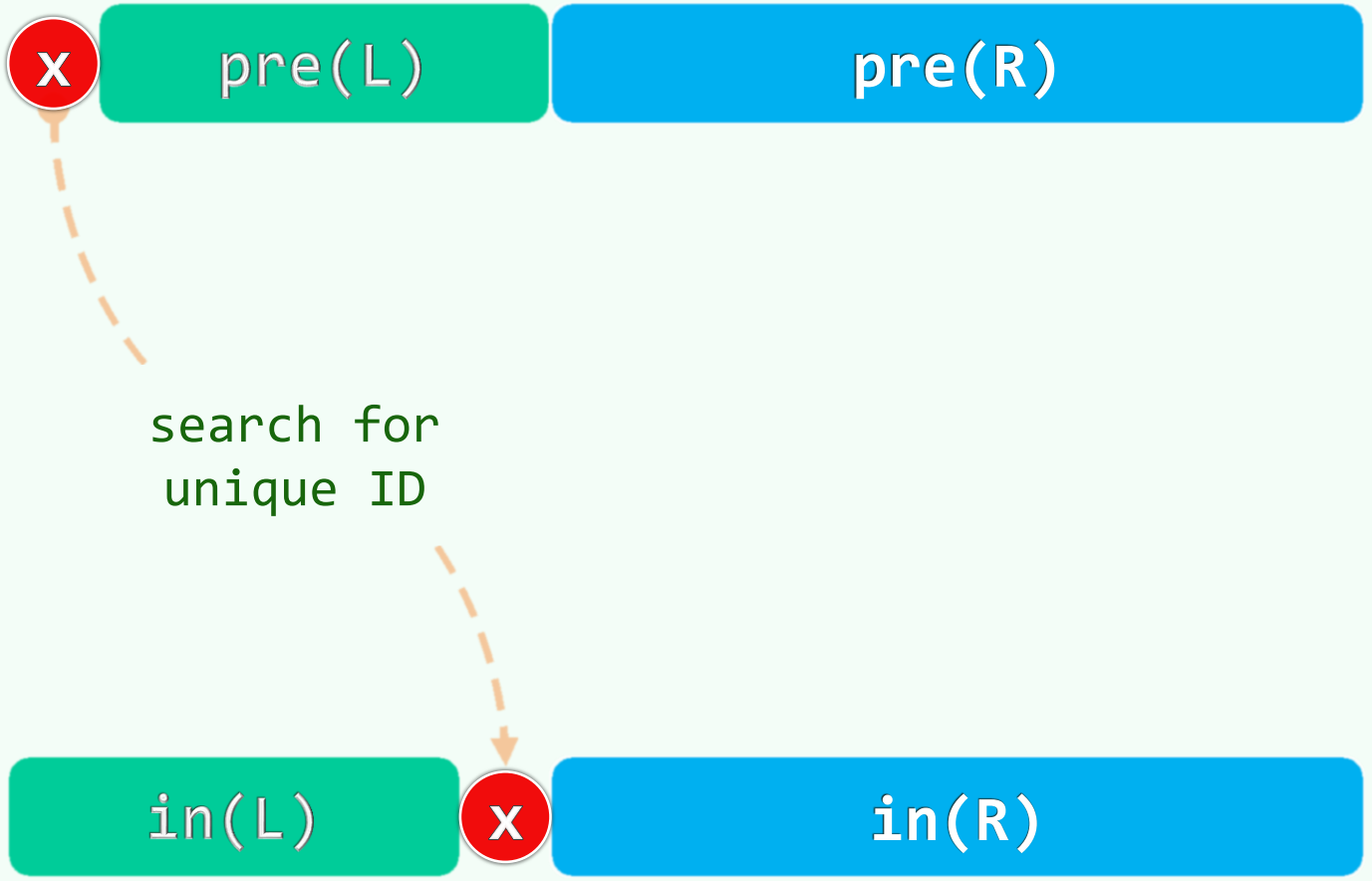
We'll find our own way back.

邓俊辉

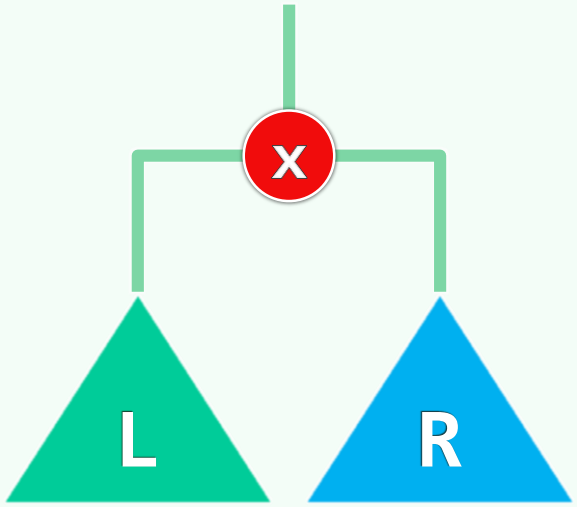
deng@tsinghua.edu.cn

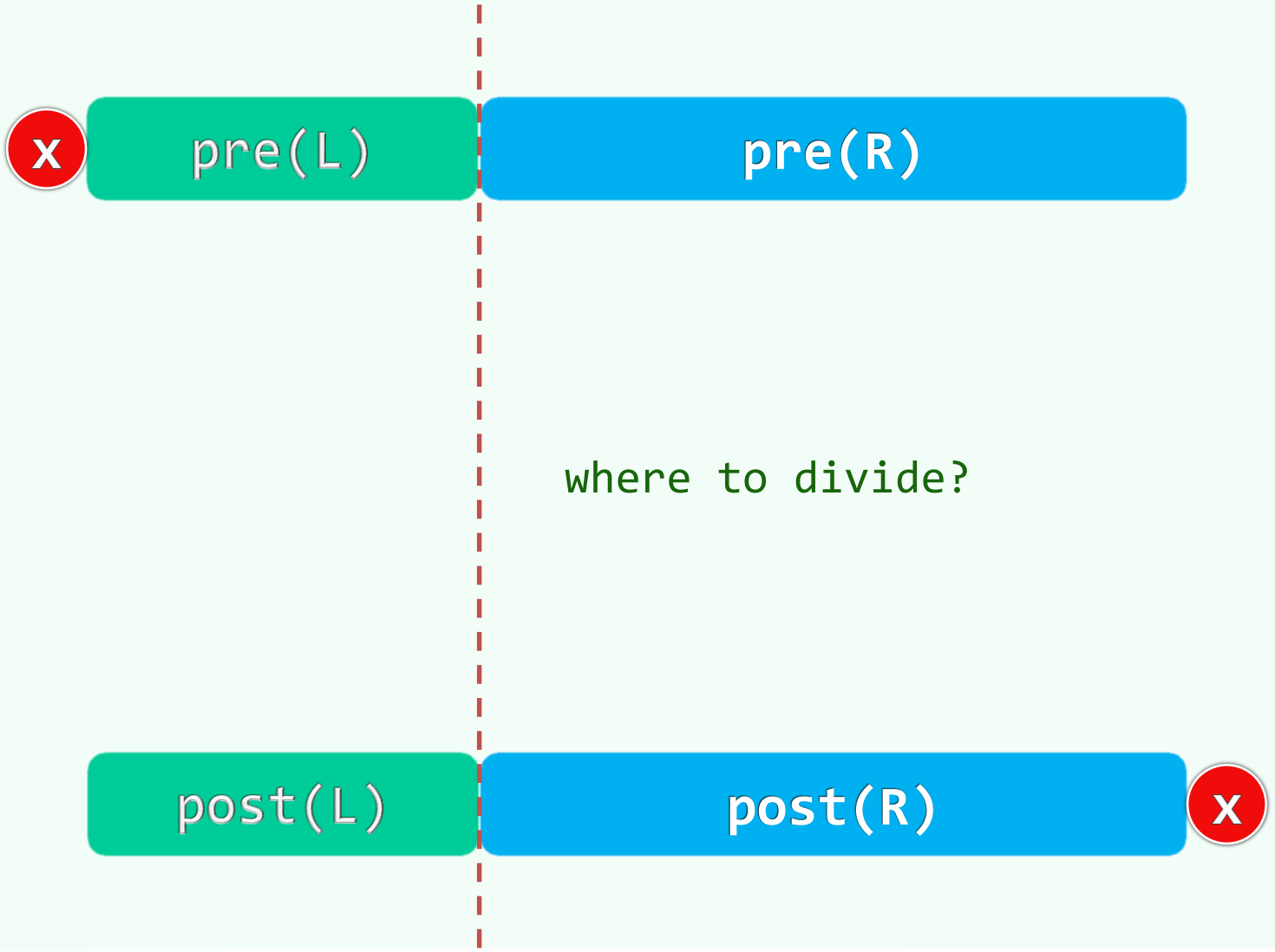
[先序 | 后序] + 中序



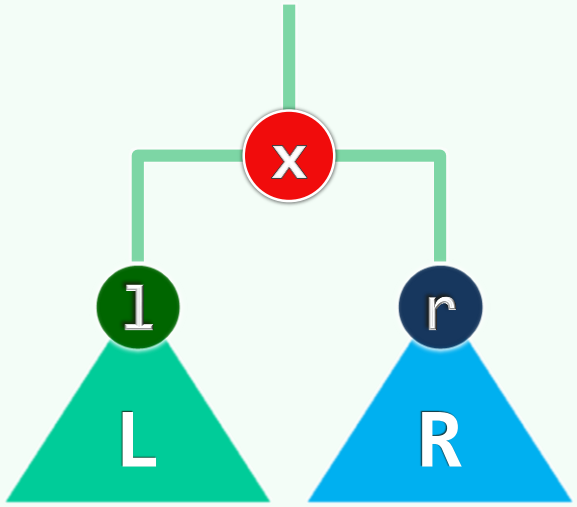


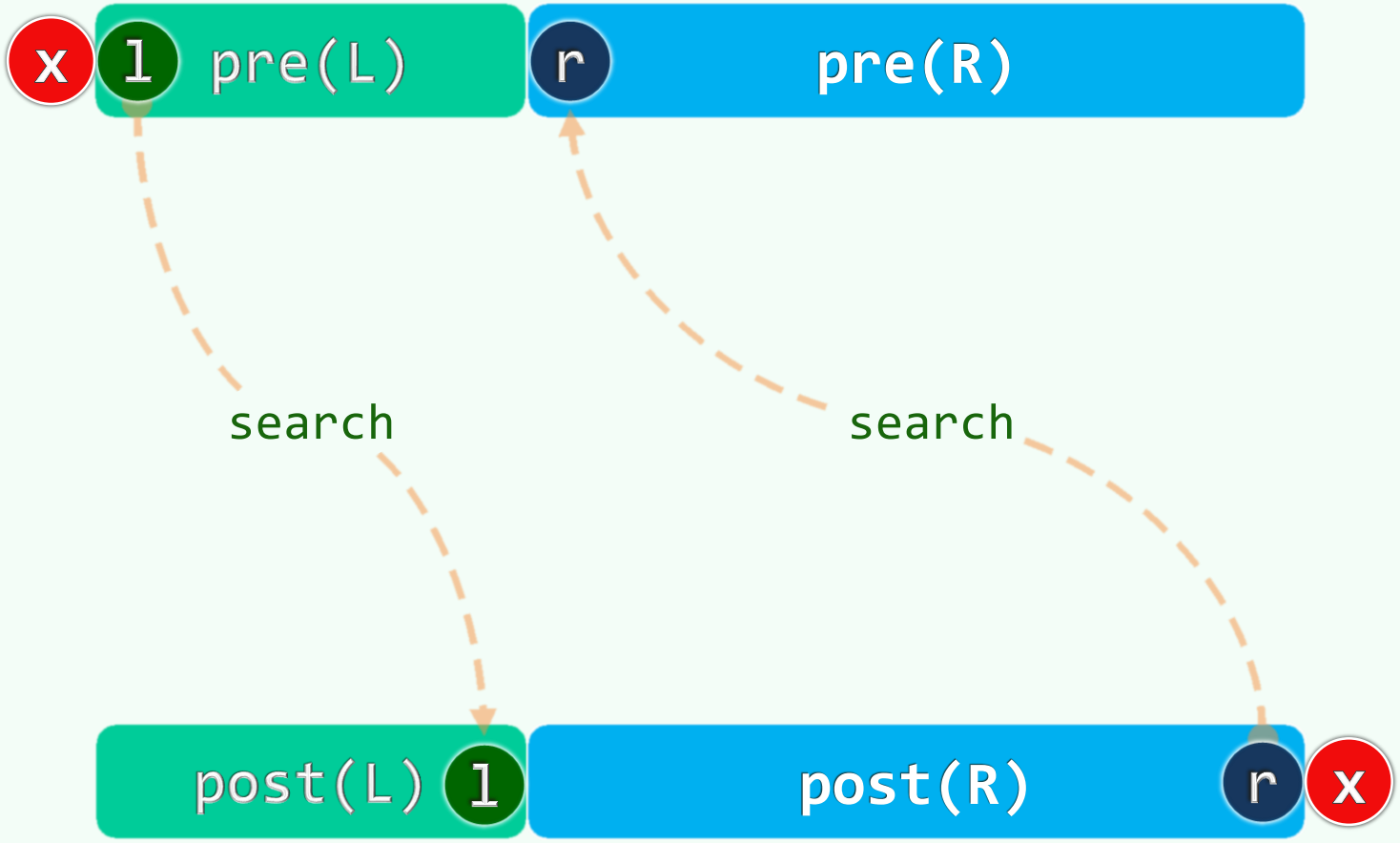
[先序 + 后序] ?



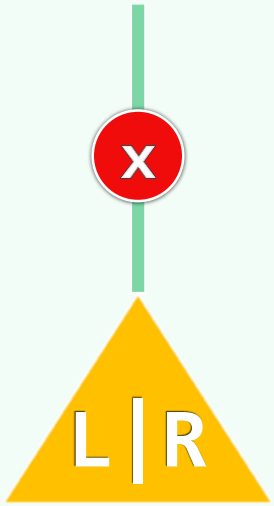


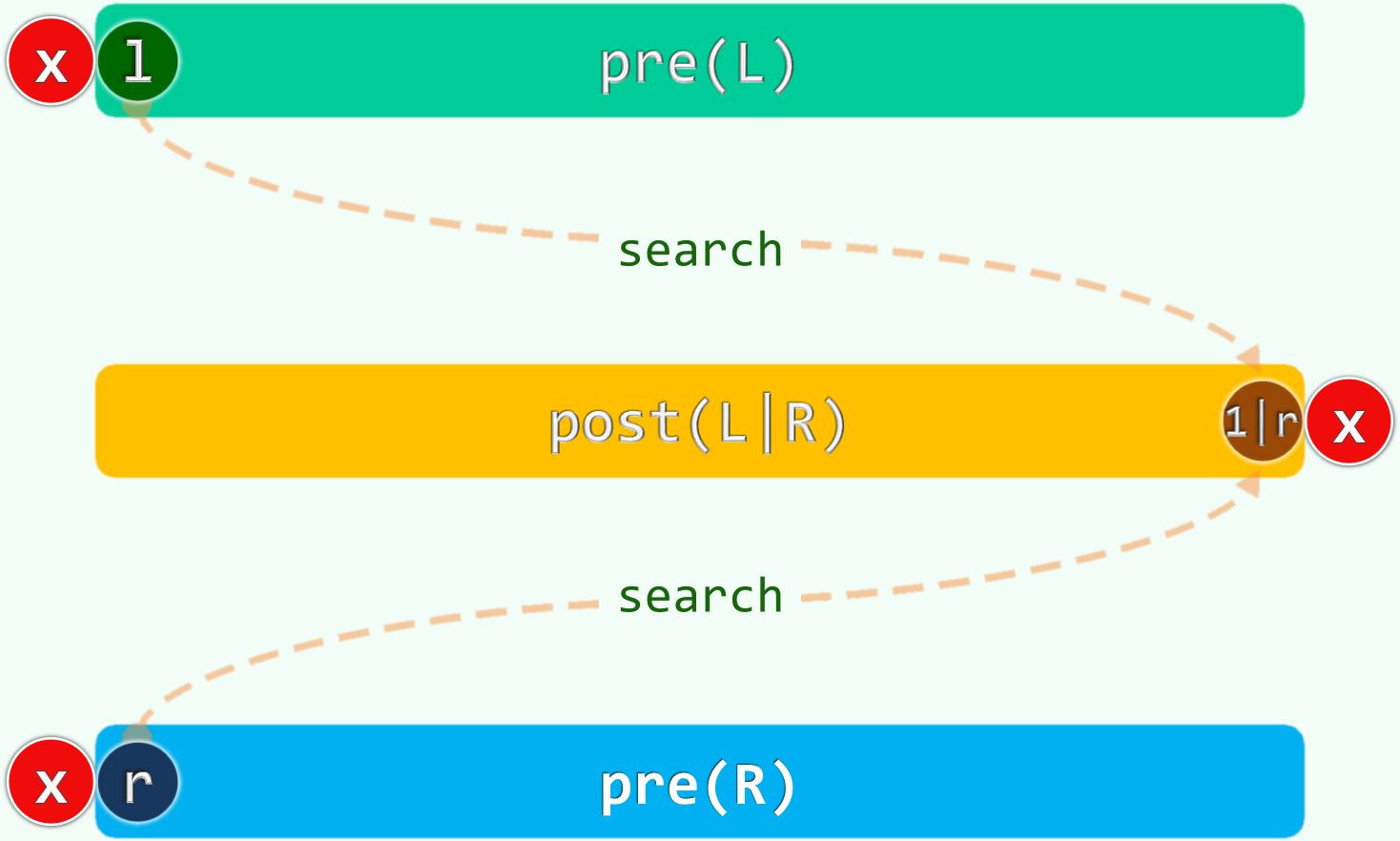
[先序 + 后序] !



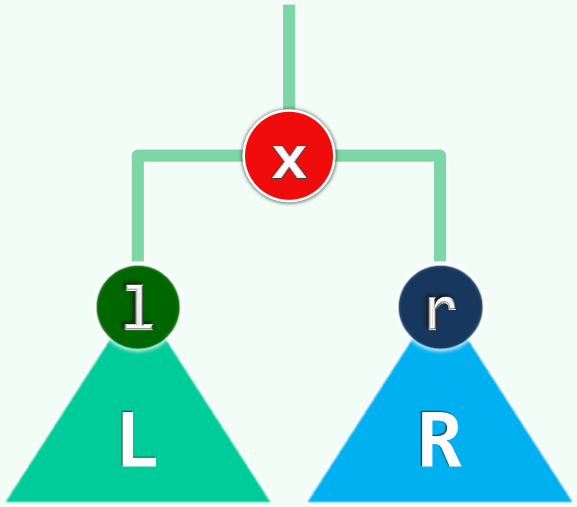


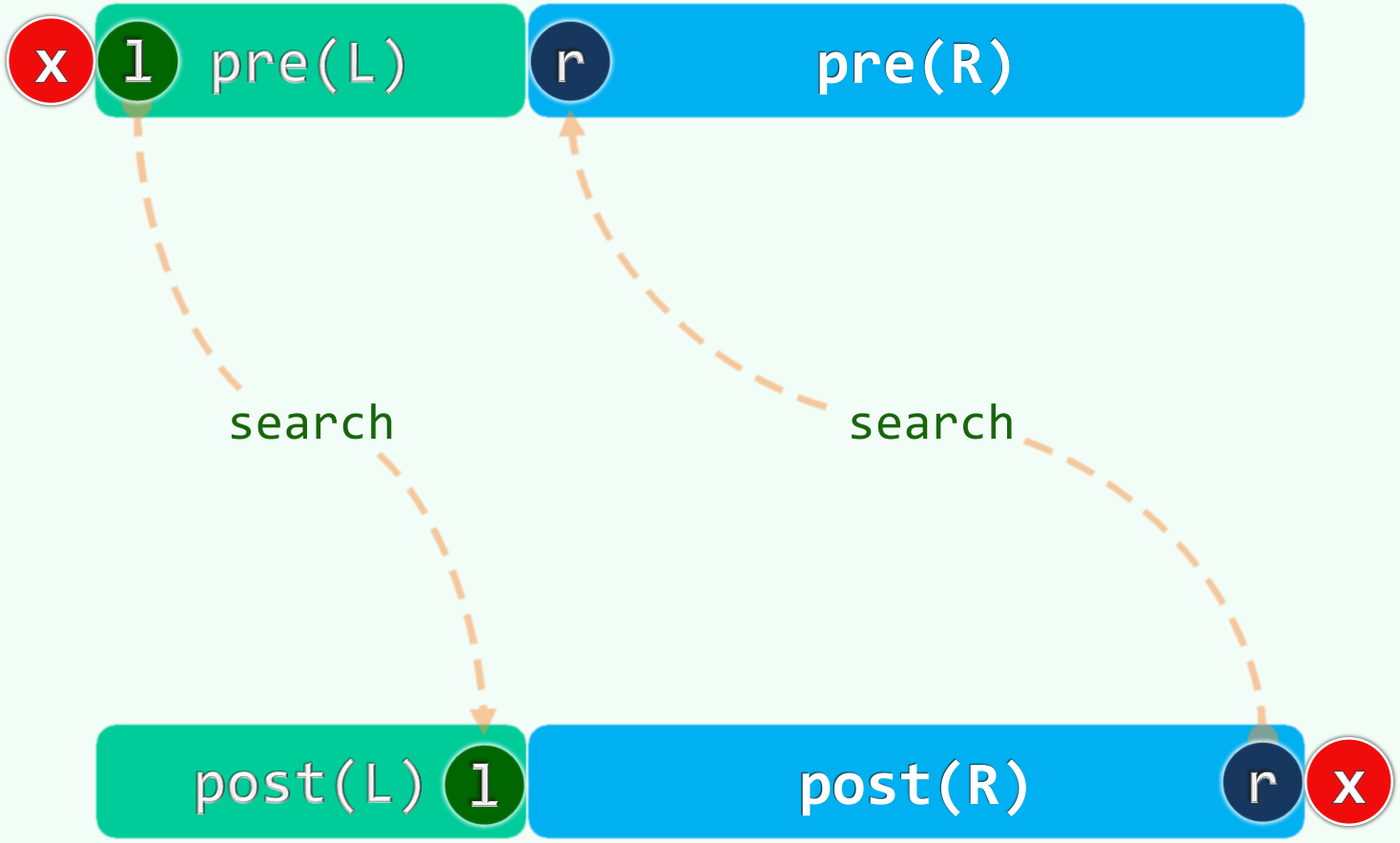
[先序 + 后序] ？？





[先序 + 后序] x 真!

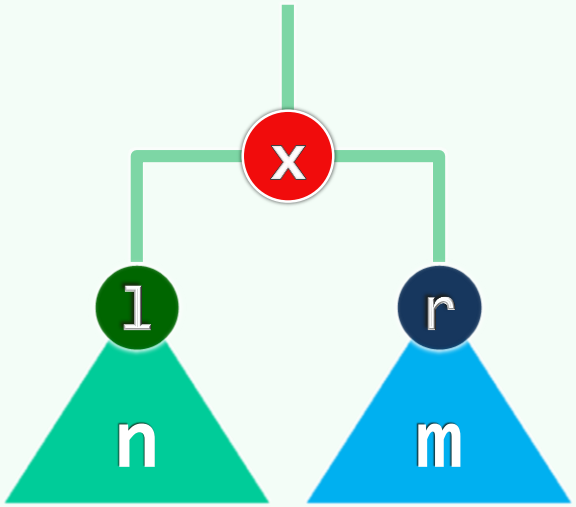




增强序列

❖ 假想地认为，每个NULL也是“真实”节点，并在遍历时一并输出

每次递归返回，同时输出一个事先约定的元字符“^”



❖ 若将遍历序列表示为一个Iterator，则可将其定义为

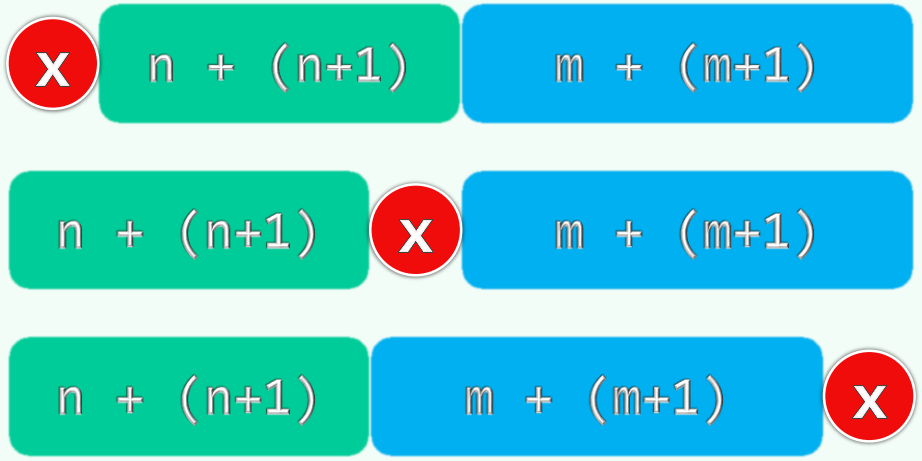
```
Vector< BinNode<T> * >
```

于是在增强的遍历序列中，这类“节点”可统一记作NULL

❖ 可归纳证明：在增强的先序、后序遍历序列中

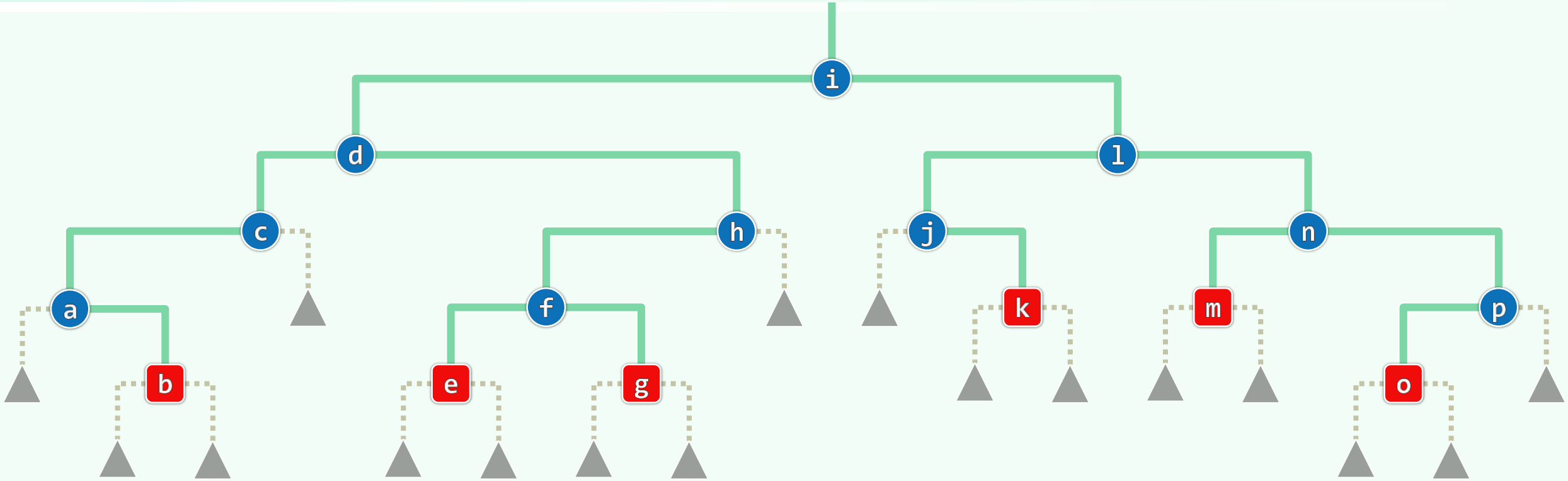
1）任一子树依然对应于一个子序列，而且

2）其中的NULL节点恰比非NULL节点多一个



❖ 如此，通过对增强序列分而治之，即可重构原树

增强序列：实例



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

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

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