

05-1

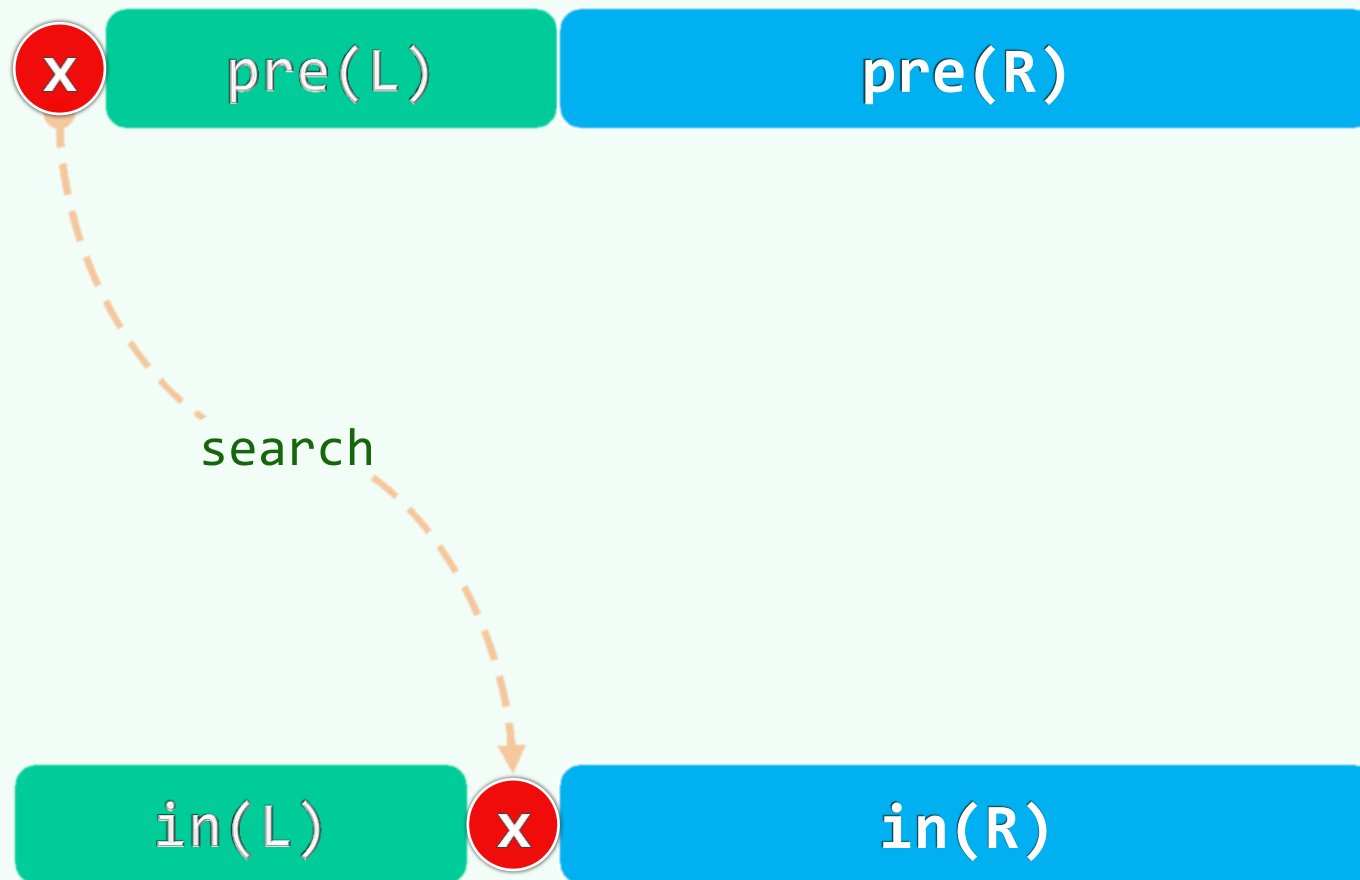
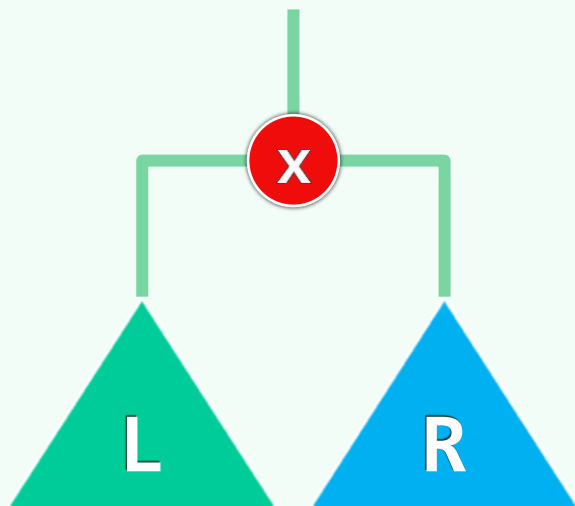
二叉树

重构

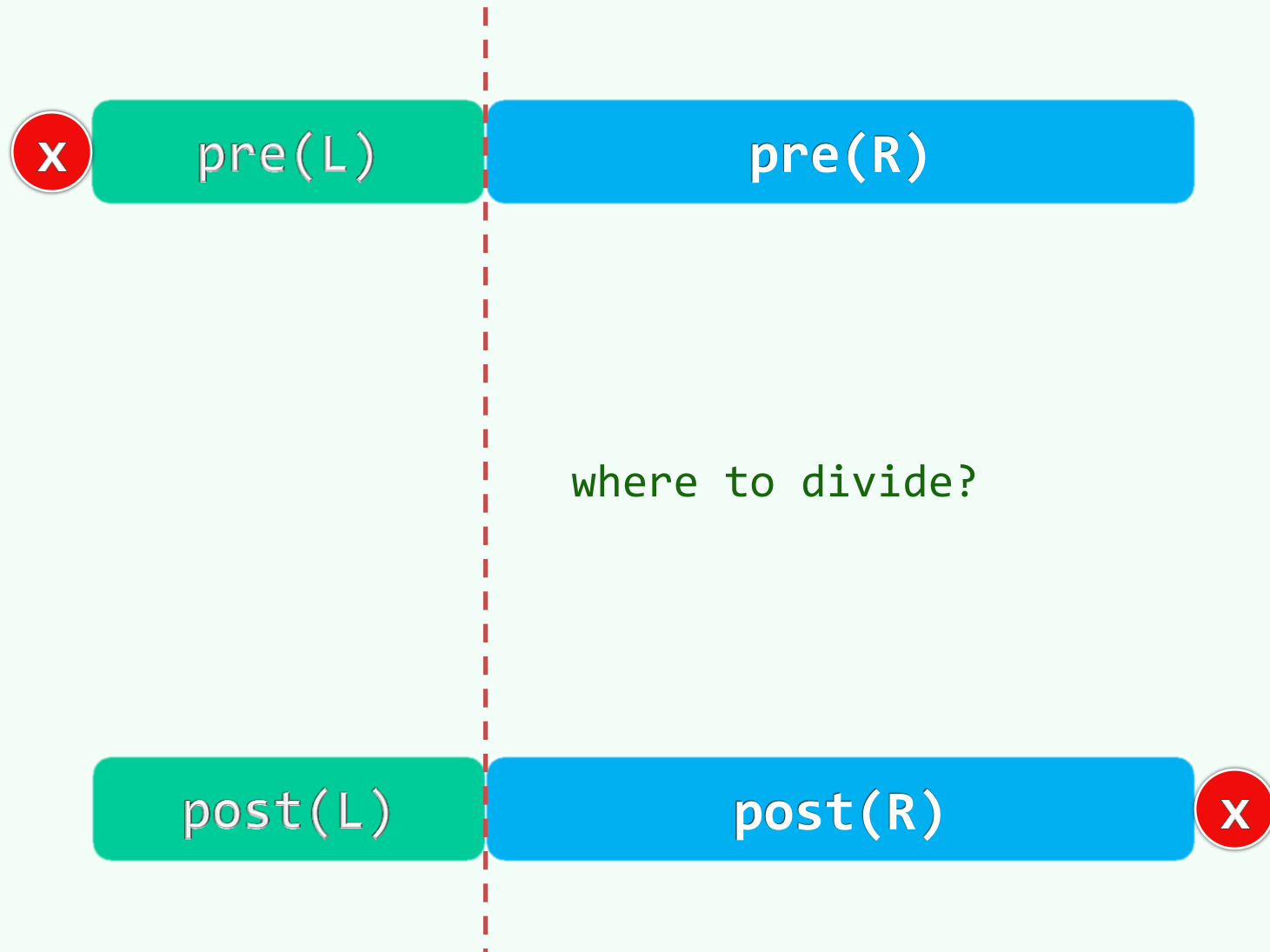
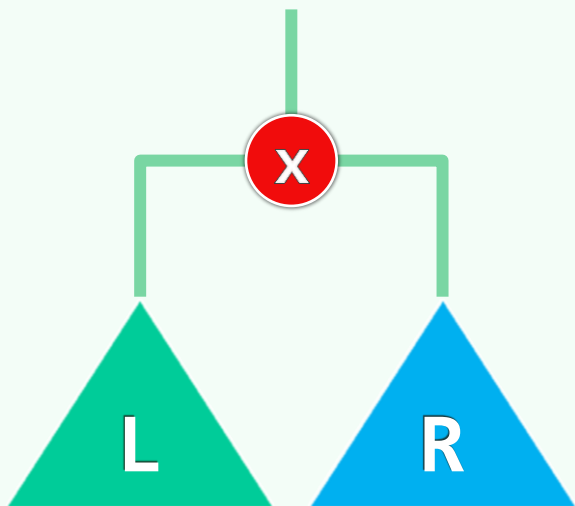
邓俊辉

deng@tsinghua.edu.cn

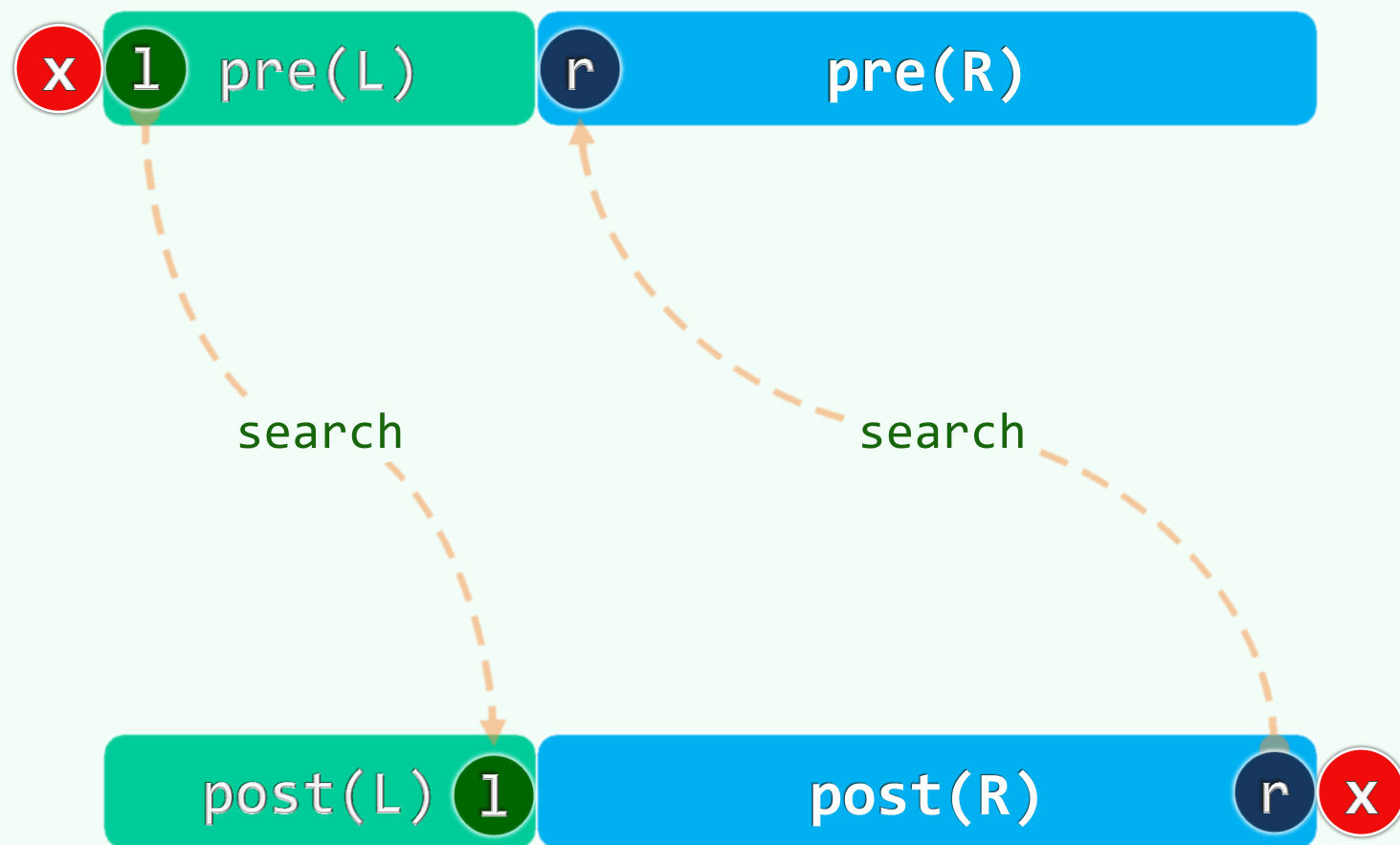
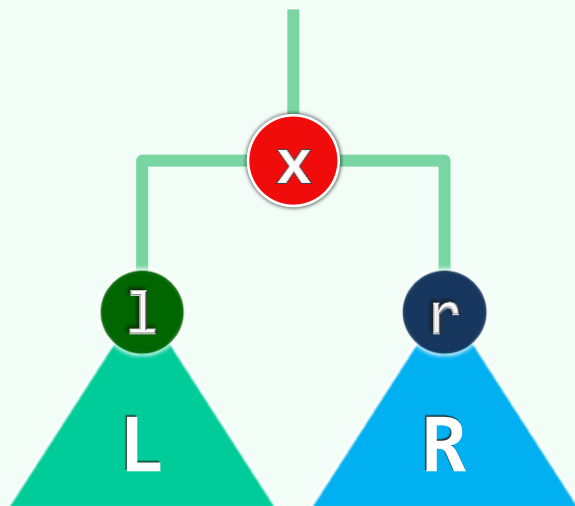
[先序 | 后序] + 中序



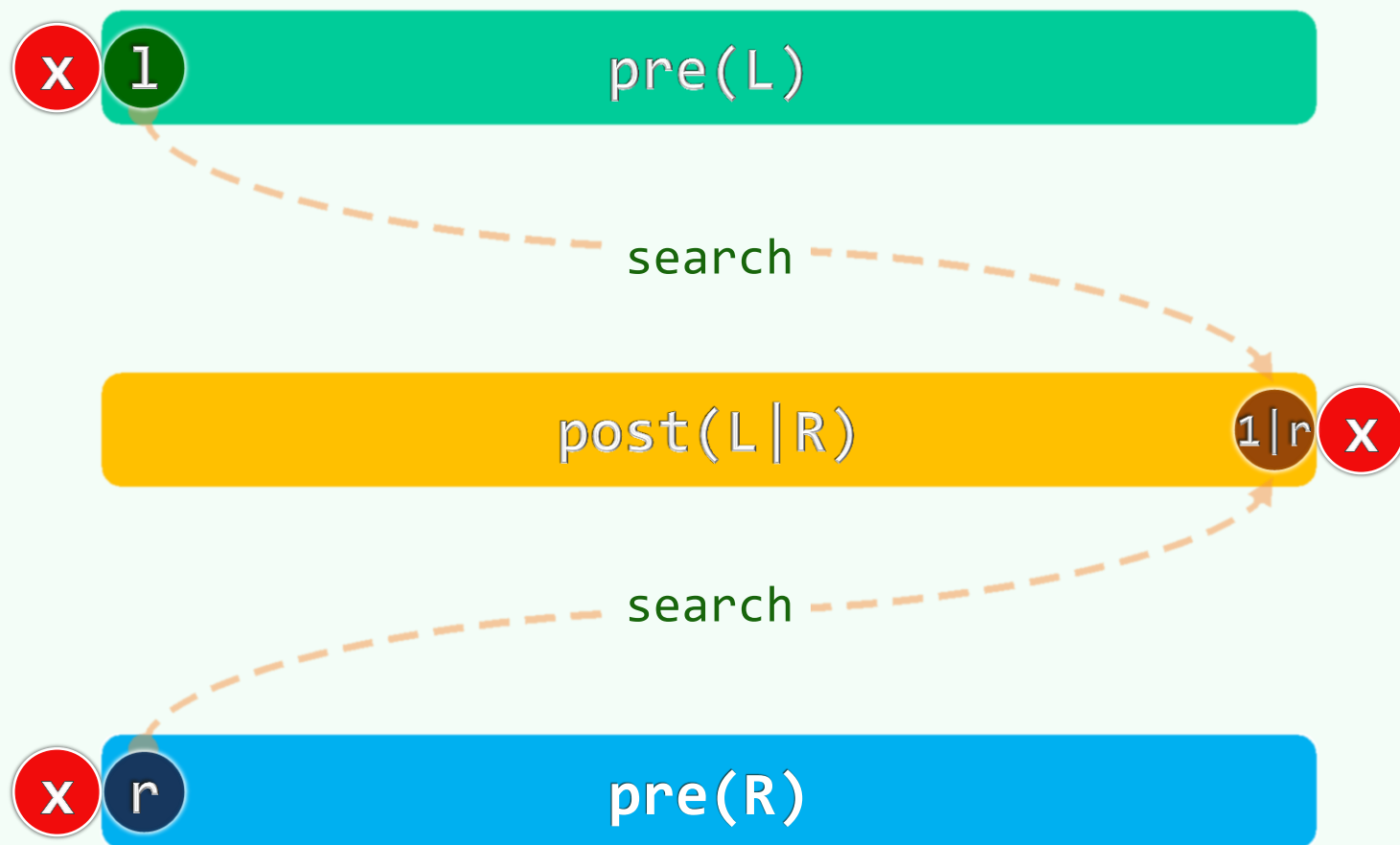
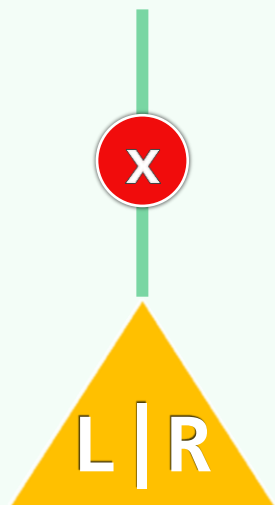
[先序 + 后序] ?



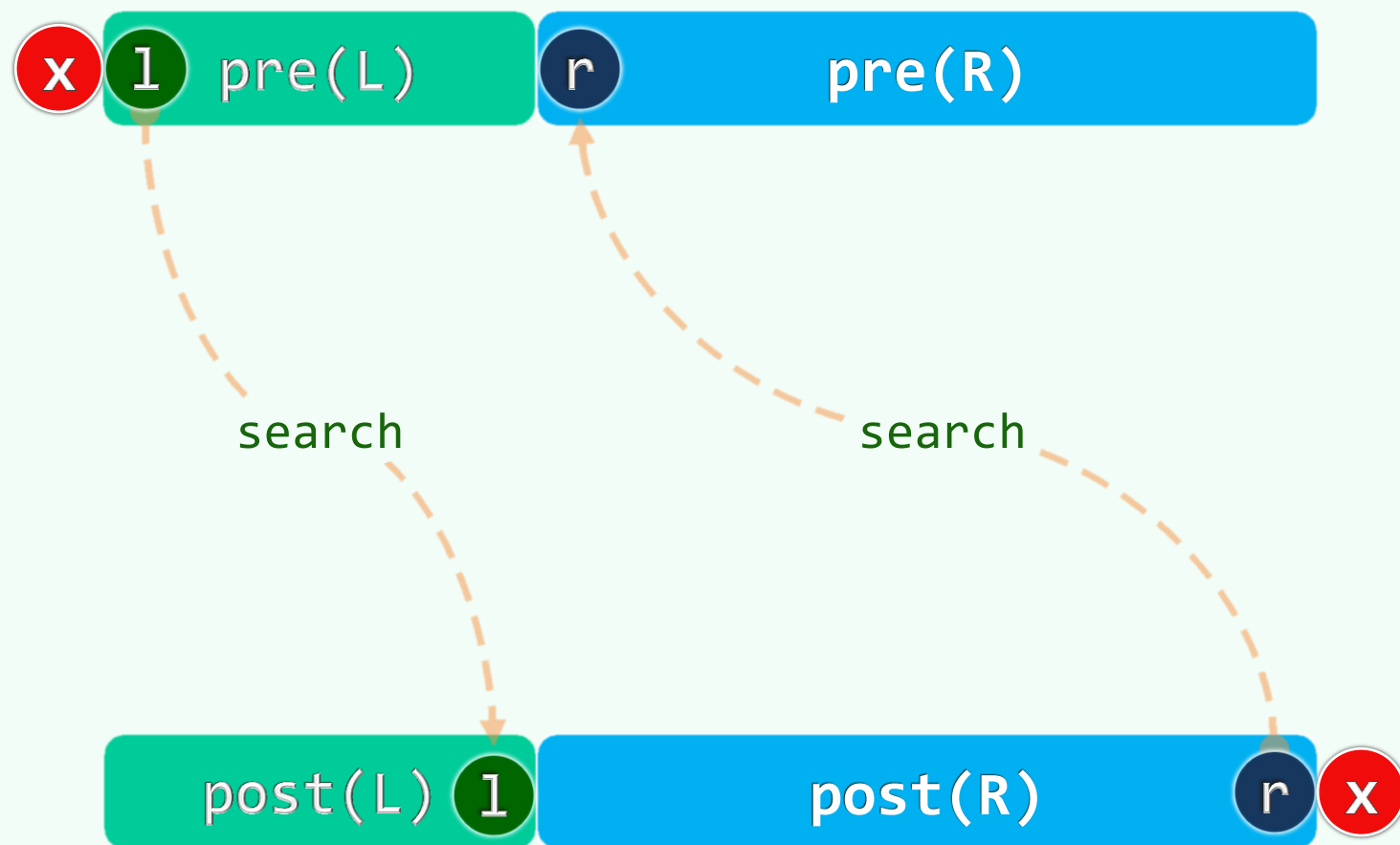
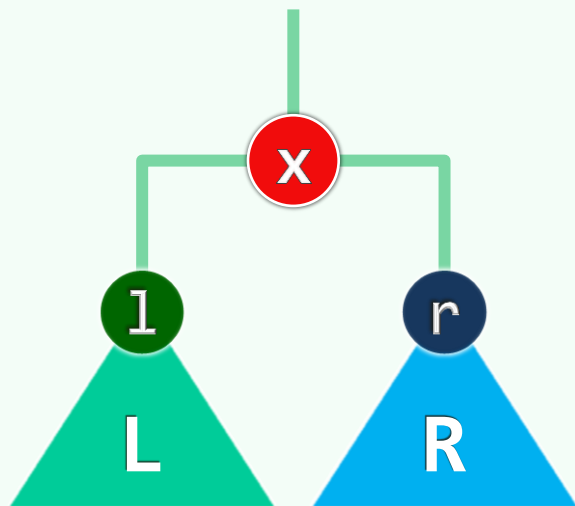
[先序 + 后序] !



[先序 + 后序] ??



[先序 + 后序] x 真!



增强序列

❖ 假想地认为，树中每个**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（比如“^”）

❖ 若将遍历序列表示为一个**Iterator**，则可将其定义为

`Vector<BinNode<T>*>`

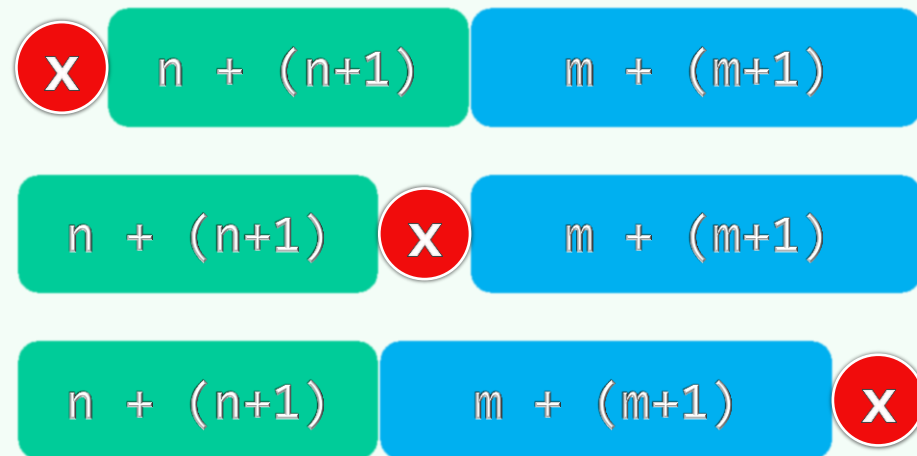
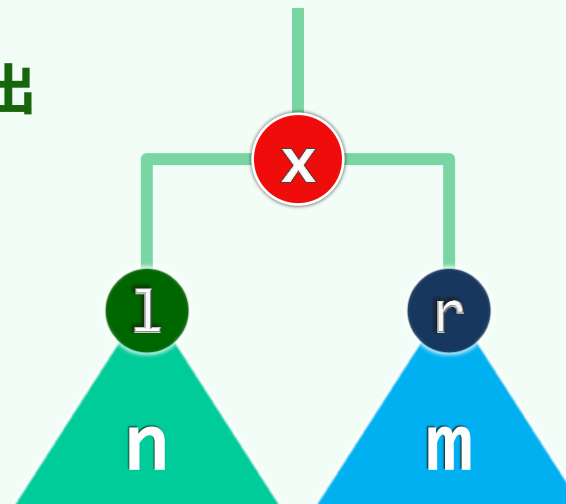
于是在**增强**的遍历序列中，这类“节点”可统一记录为**NULL**

❖ 可归纳证明：在增强的先序、中序、后序遍历序列中

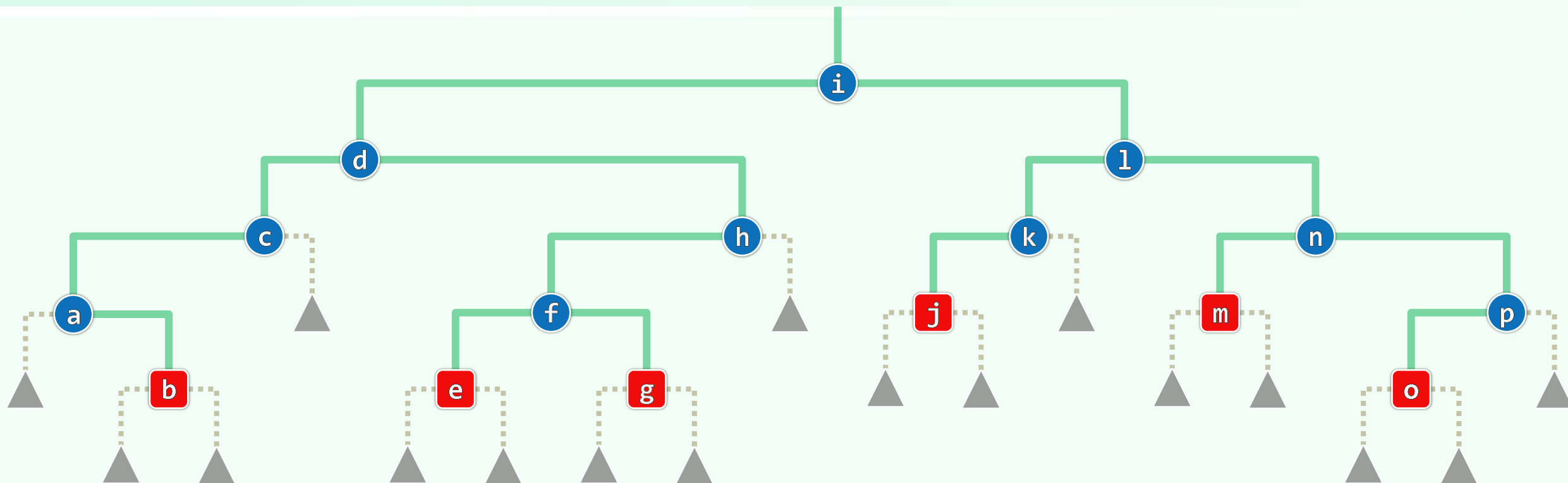
1) 任一子树依然对应于一个子序列，而且

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 k j ^ ^ ^ 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 ^ ^ j ^ k ^ ^ m ^ ^ o ^ p n l i