07-D1

搜索树应用

多层搜索树：一维

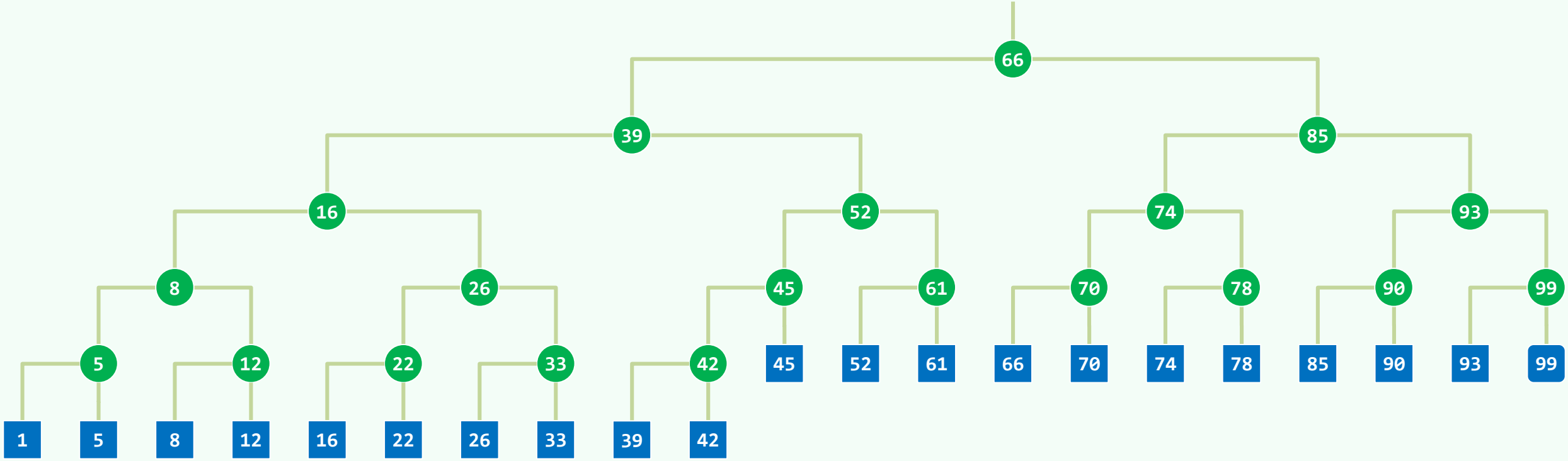
邓俊辉

deng@tsinghua.edu.cn

只在此山中，云深不知处

完全的二叉搜索树

- $\forall v,\ v.key = \min\{\, u.key \mid u \in v.rTree \,\} = v.succ.key$

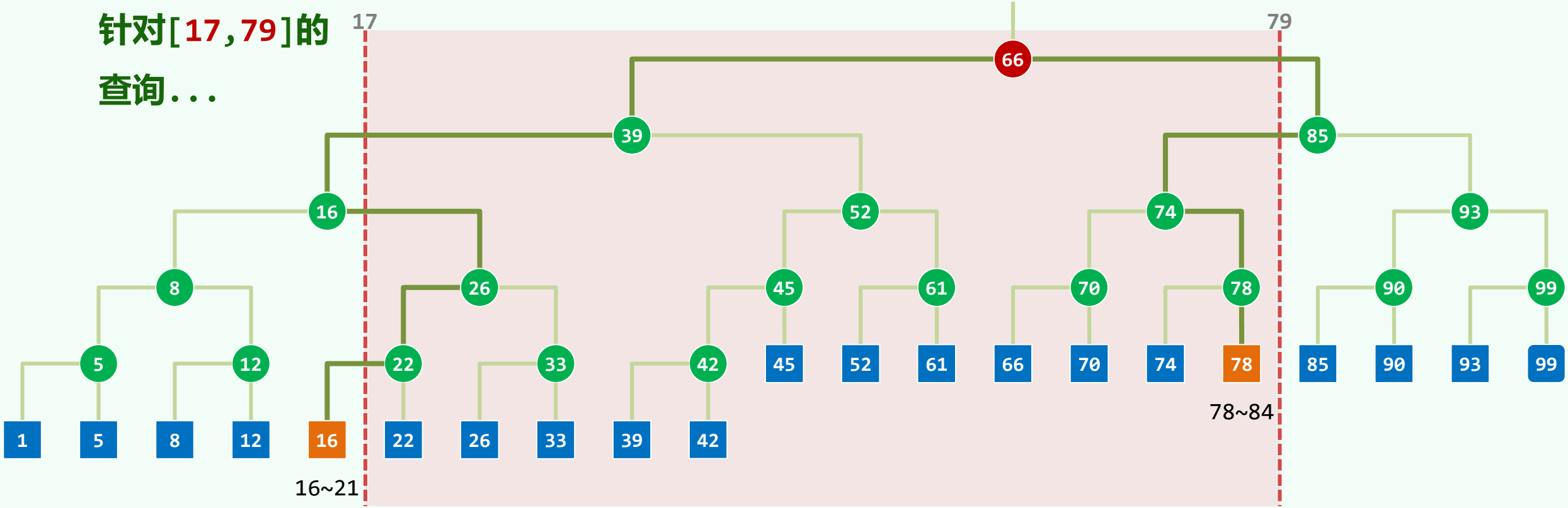


$\forall u \in v.lTree,\ u.key < v.key$ $\forall u \in v.rTree,\ u.key \geq v.key$

- search(x) ： 返回不超过x的最大者

最低公共祖先

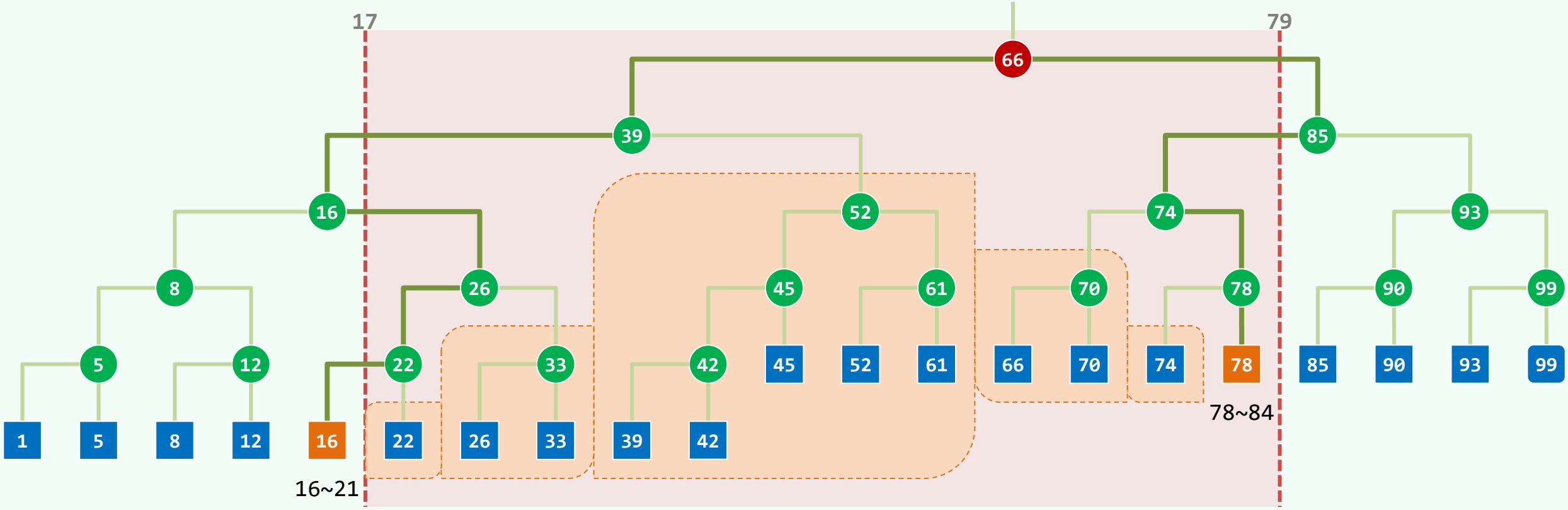
- 作为实例，考查针对[17,79]的查询...



- 两次搜索：search(17) = 16（可能排除） + search(79) = 78（必然接受）
- 考查LCA(16,78) = 66 //Lowest Common Ancester

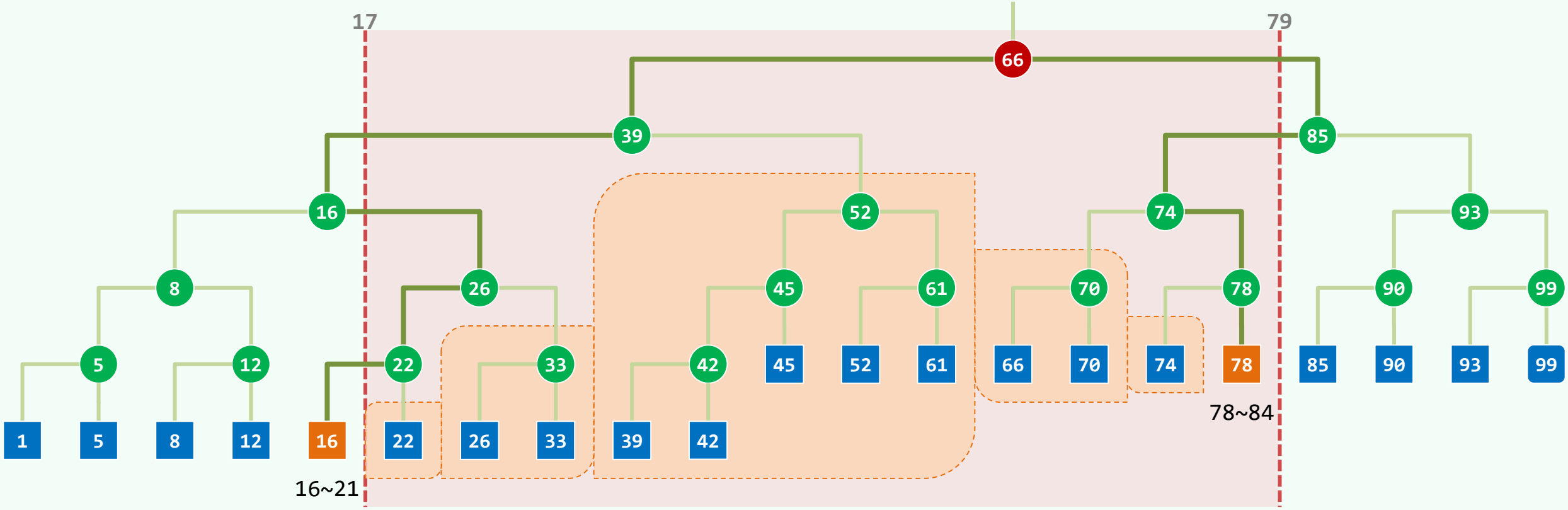
查询结果 = O(logn)棵彼此无交的子树

- 从LCA出发，分别检视path(16)和path(78)



- 沿着path(16)/path(78)：所有的右转/左转均可忽略，而每棵右子树/左子树则直接报告

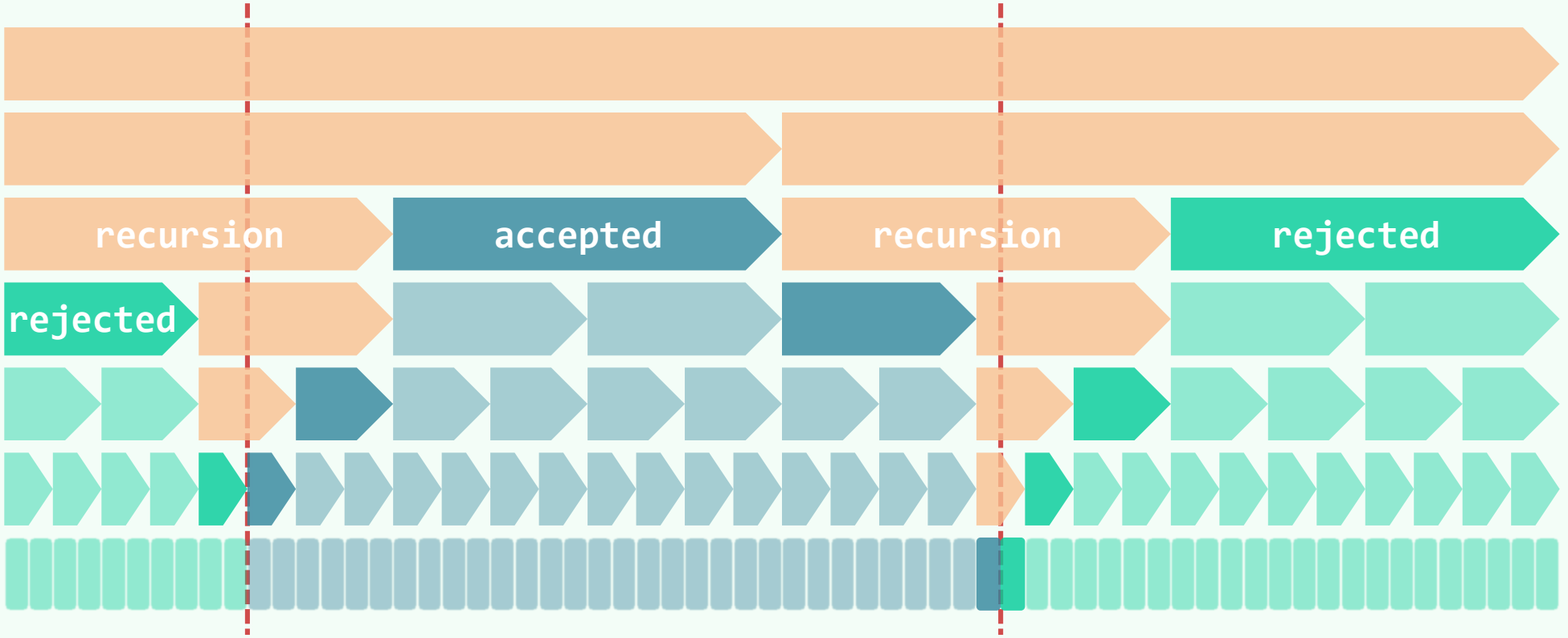
复杂度



查询： $\mathcal{O}(\log n)$ **预处理：** $\mathcal{O}(n \log n)$ **存储：** $\mathcal{O}(n)$

热刀来切O(logn)层的巧克力蛋糕

- Region(u)被Region(v)包含，当且仅当节点u是v的后代
- Region(u)与Region(v)无交，当且仅当u和v不是直系血缘关系



- 所谓的查询，无非是将节点分为三类：accepted + rejected + recursion