

作业6

Exercise1

- 首先考虑覆盖：若 $u \leq x, v \leq y$ ，就说明点 (u, v) 被点 (x, y) 覆盖
- 预处理方法是对于每个点，记下： $n(x, y) = |(0, x] \times (0, y] \cap P|$
- 为此需要花费 $O(n^2)$ 的时间在预处理阶段
- 在查找阶段考虑容斥原理。要找一个区域里面的点，需要考虑这个区域的左下角、左上角、右上角、右下角的四个点的覆盖的数量

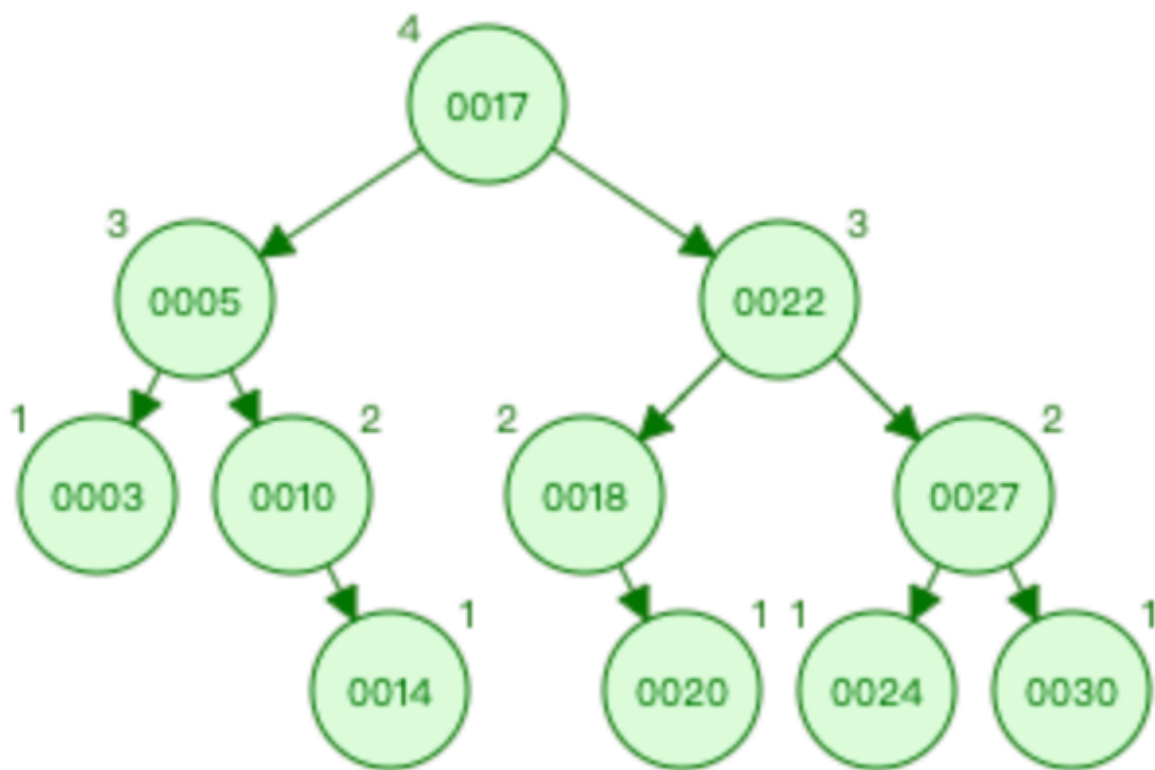
❖ 于是对任意的 $R = (x_1, x_2] \times (y_1, y_2]$ ，有

$$|R \cap P| = n(X_2, Y_2) - n(X_1, Y_2) - n(X_2, Y_1) + n(X_1, Y_1)$$

- 这样效果就非常快，只需要 $O(\log n)$ 的时间，但是需要 n^2 的空间

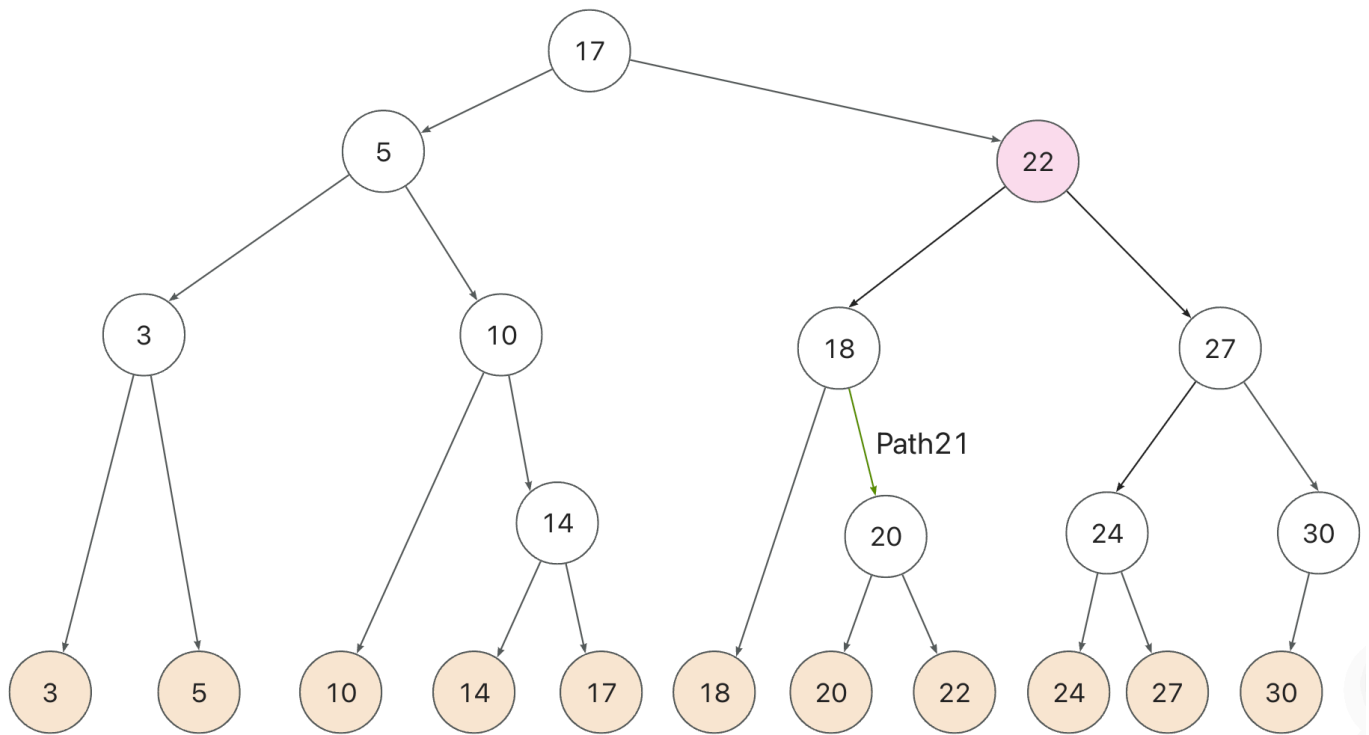
Exercise2

- 如下树所示



- 先搜索21，搜索的结果终止于20号，这是path(21)
- 再搜索27，搜索的结果终止与27号，这是path(27)

- 可以得到两个路径的他们的共同祖先是22
- 从共同祖先是22的左儿子出发，然后沿着path(21)，忽略右拐，一旦左拐就输出右子树，对于终点单独检查发现20不满足条件。最终发现没有输出
- 从共同祖先是22的右儿子出发，然后沿path(27)，忽略左拐，一旦右拐输出左子树（结果22的右儿子就是27，相当于没有走）对于27这个节点，发现他还有左儿子，就执行输出，输出了（24、27）
- 再考虑共同的祖先节点22（这个需要单独加进来）
- 所以一共是22、24、27
- 个人认为PPT的算法说的不严谨，需要有所修正



- PPT说的不严谨的地方是一旦左拐（从共同祖先出发的时候左拐是不算的，否则就会输出错误的东西）

BBST: 查找

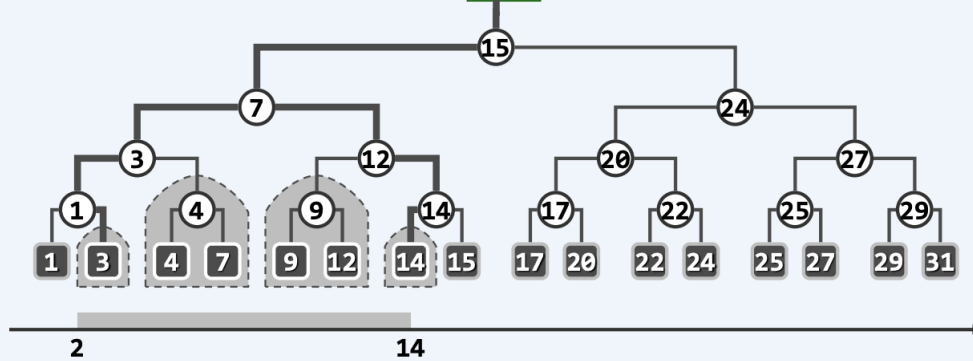
❖ 分别查找2和14, 终止于3和14

//其最低共同祖先LCA(3, 14) = 7

❖ 从LCA起向下, 重走一遍path(3)和path(14)

沿path(3)/path(14), 忽略右/左拐, 一旦左/右拐则输出右/左子树

❖ 最后, 还要单独检查path(3)和path(14)的终点

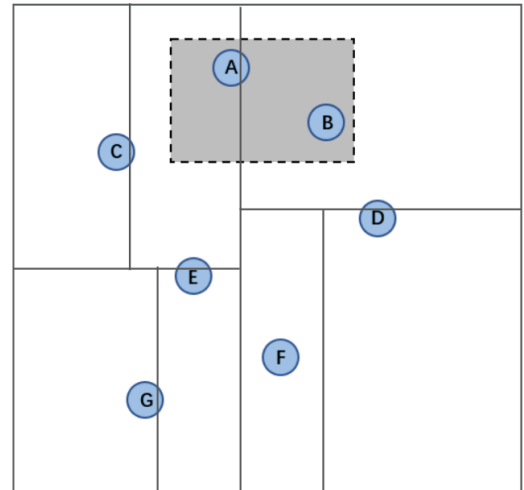
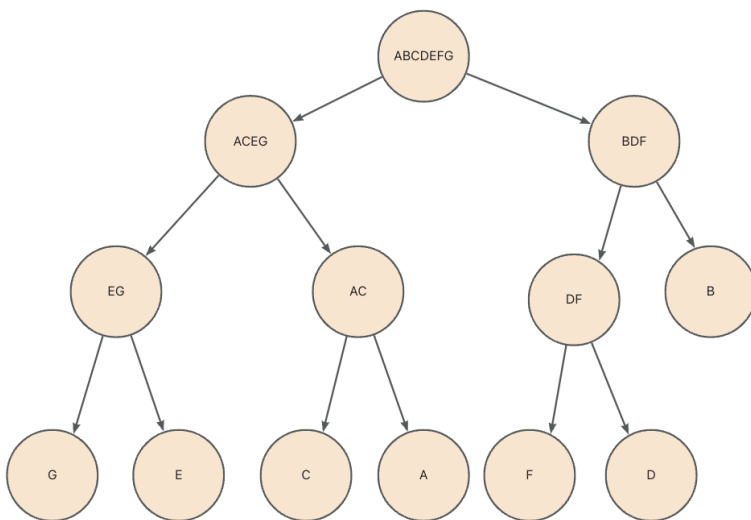


Data Structures (Spring 2015), Tsinghua University

12

Exercise3

- 画图结果如下所示



Exercise4

- 发现搜索的区域同时包含了两个根节点儿子的区域（也就是说对于根节点ABCDEFG的左右两个儿子的范围都有覆盖）
 - 在ACEG节点递归搜索灰色的部分的左边区域（简称左灰），发现左灰部分完全在ACEG的右儿子（AC）区域部分。进一步递归搜索。
 - 在AC节点递归搜索左灰部分，发现左灰区域完全在AC节点的右儿子区域里面
 - 搜索发现了A节点

- 在BDF节点递归搜索灰色的部分的右边区域（简称右灰），发现右灰部分完全在BDF节点的右儿子范围里面
- 递归进入B节点，发现了B节点
- 搜索结束，发现了AB节点

Exercise5

消除多点共垂直、共水平的退化情况？

- 轮换坐标轴，比如我们上面Exercise3采用就是这种轮流用x、y轴来划分。其实如果不严格要求使用什么轴来划分区域，我们尽可能选择不那么容易出现重复的轴
 - 比如假设x轴坐标容易重复，那我就尽可能用y轴划分
 - 比如假设y轴坐标容易重复，那我就尽可能用x轴划分
 - 数据分析中，通常选择方差更大的作为轴
- 从根源解决问题，尽可能避免数据的重复，假如我的数据能精确到小数点后3位，并且这时候能保证数据各不相同，那我就尽可能的保留三位小数，而不是保留整数（带来更多的数字重复性问题），来构造这个KD树
- 添加随机扰动：如果可以对于数据做一个适当范围的微小修改，可以在每一次划分数据集时，可以添加一个随机扰动，使得数据点稍微偏离垂直或水平的线。这样可以避免多点共垂直或共水平的情况，从而避免 KD-Tree 的退化。