8.高级搜索树

(xb1) kd-树：一维

邓俊辉

deng@tsinghua.edu.cn

范围查找

❖ 在直线L上，给定点集P = { p_0, …, p_{n-1} }

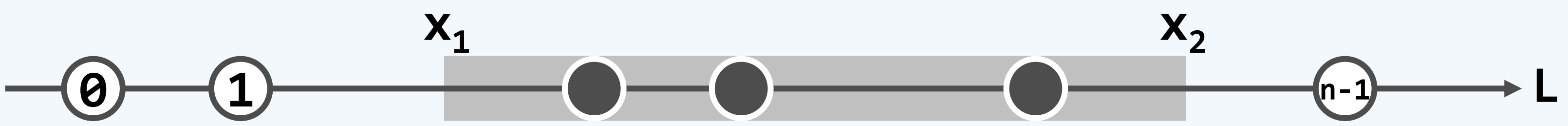
❖ 在任意区间R = [x_1, x_2]内： 有多少点 counting ？ 有哪些点 reporting ？

❖ 限制：点集规模 n 非常大，以致于需要借助 外存

因此：通过遍历整个点集， 逐一测试 各点将非常耗时，应尽量避免

❖ 问题特点： Offline P相对固定，可以 离线 方式预处理

Online R数量巨大，以非确定方式在线逐个给出，须 在线 处理



蛮力法

❖ 依次检查P中的每个点：若位于指定区间之内，则计数（或将其加至查找结果中）

❖ $\Theta(n)$时间——能否更快？就渐进意义而言似乎不能，因为...

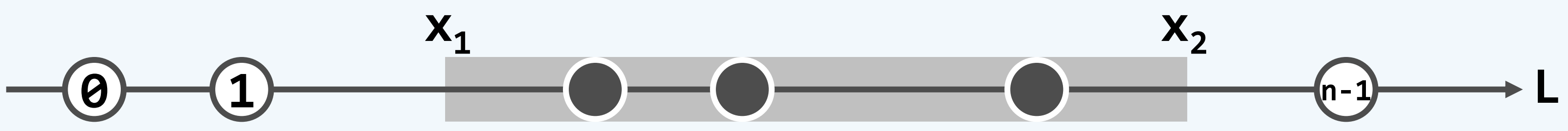
❖ 最坏情况下，命中的点数 $\boxed{r = \Omega(n)}$ ——即便直接输出它们，也要花费 $\boxed{\Omega(n)}$ 时间

❖ 实际上，相对于查找过程本身，输出过程的效率不甚重要

对于简单的Counting版，甚至不必输出命中子集

蛮力查找过程需反复I/O，$\Theta(n)$的常系数很大

因此，理应首先优化查找过程...



计数法

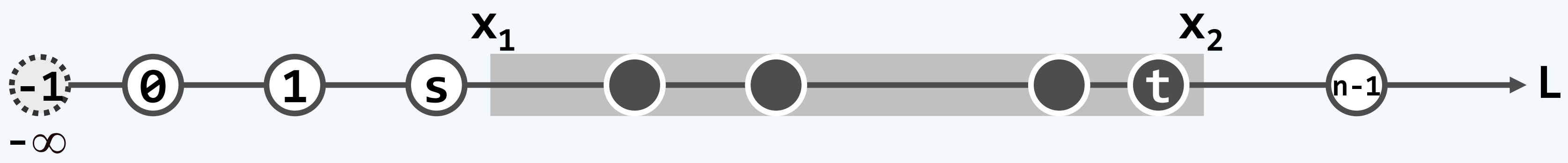
❖ 预处理：点集排序后转换为 $\boxed{\text{有序向量}}$（哨兵 $p_{-1} = -\infty$） //$\mathcal{O}(n\log n)$

❖ 查找：针对任意区间 $\boxed{R = [x_1,\ x_2]}$

1. $\boxed{t} = search(x_2) = \max\{\ i \mid x_2 \geq p_i\ \}$ //$\mathcal{O}(\log n)$

2. 从 $\boxed{p_t}$ 出发，自右向左检查各点，直至首次离开查询区间的 $\boxed{p_s}$

   只要当前点在范围之内，则报告之 //$\mathcal{O}(r + 1) = \mathcal{O}(t - s + 1)$



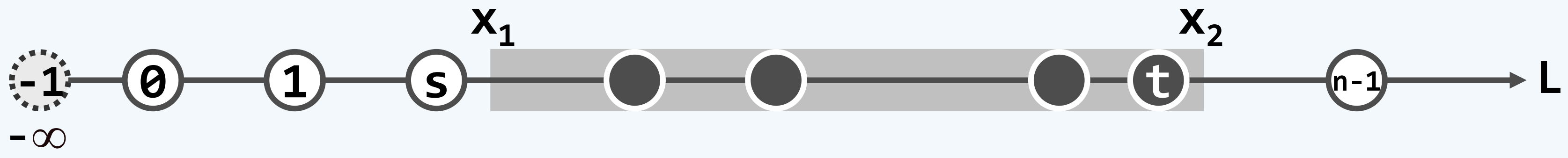
计数法

❖ 优势：$\boxed{\mathcal{O}(r + \log n)}$，输出敏感

仅涉及$\boxed{r + 1}$个点，且它们依次毗邻，I/O大大节省

对于Counting版，甚至还可进一步优化至$\boxed{\mathcal{O}(\log n)}$

⌛ 这个方法似乎不错，但是...



平面版本

- 给定平面点集

  $P = \{ p_0, \ldots, p_{n-1} \}$

- 矩形范围查找 Rectangular Range Search

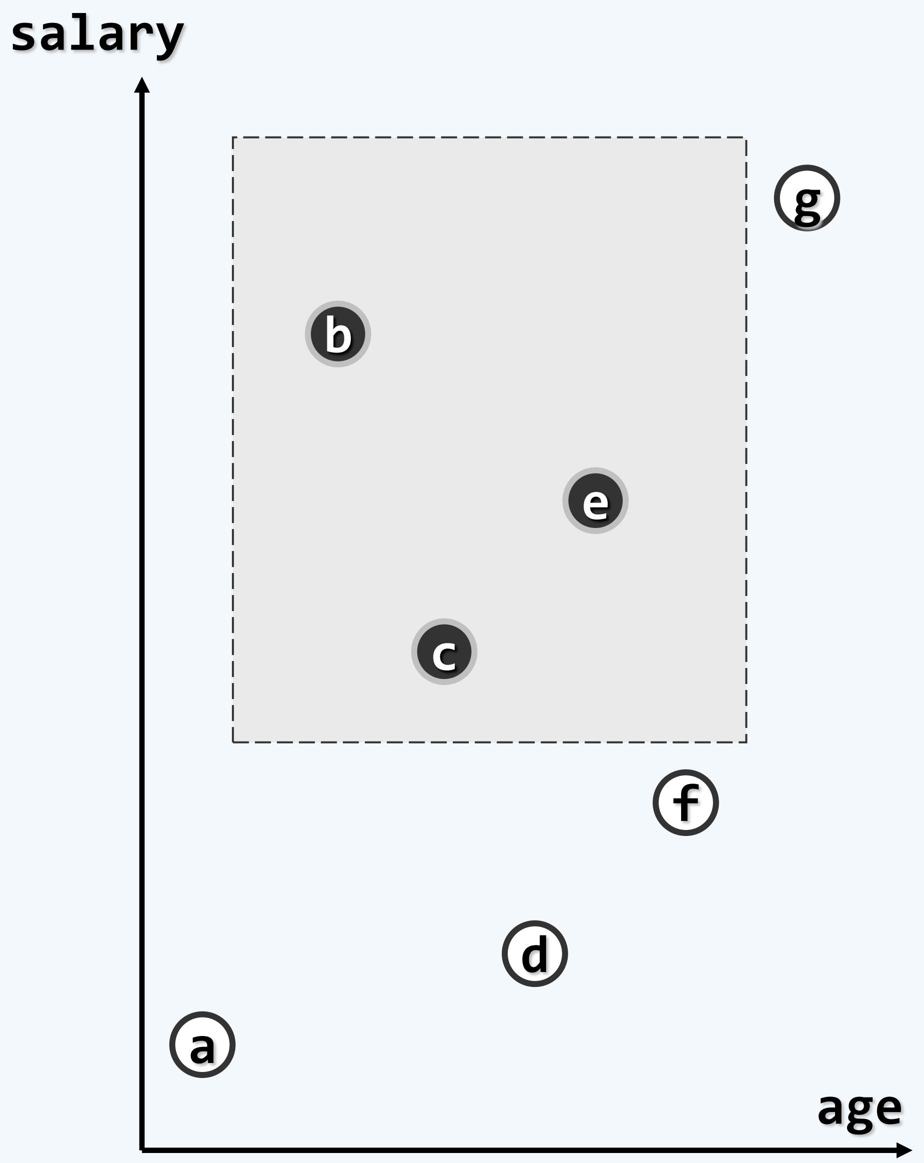
  任意矩形 $R = [x_1, x_2] \times [y_1, y_2]$ 内

  Counting 有多少个点

  Reporting 有哪些点

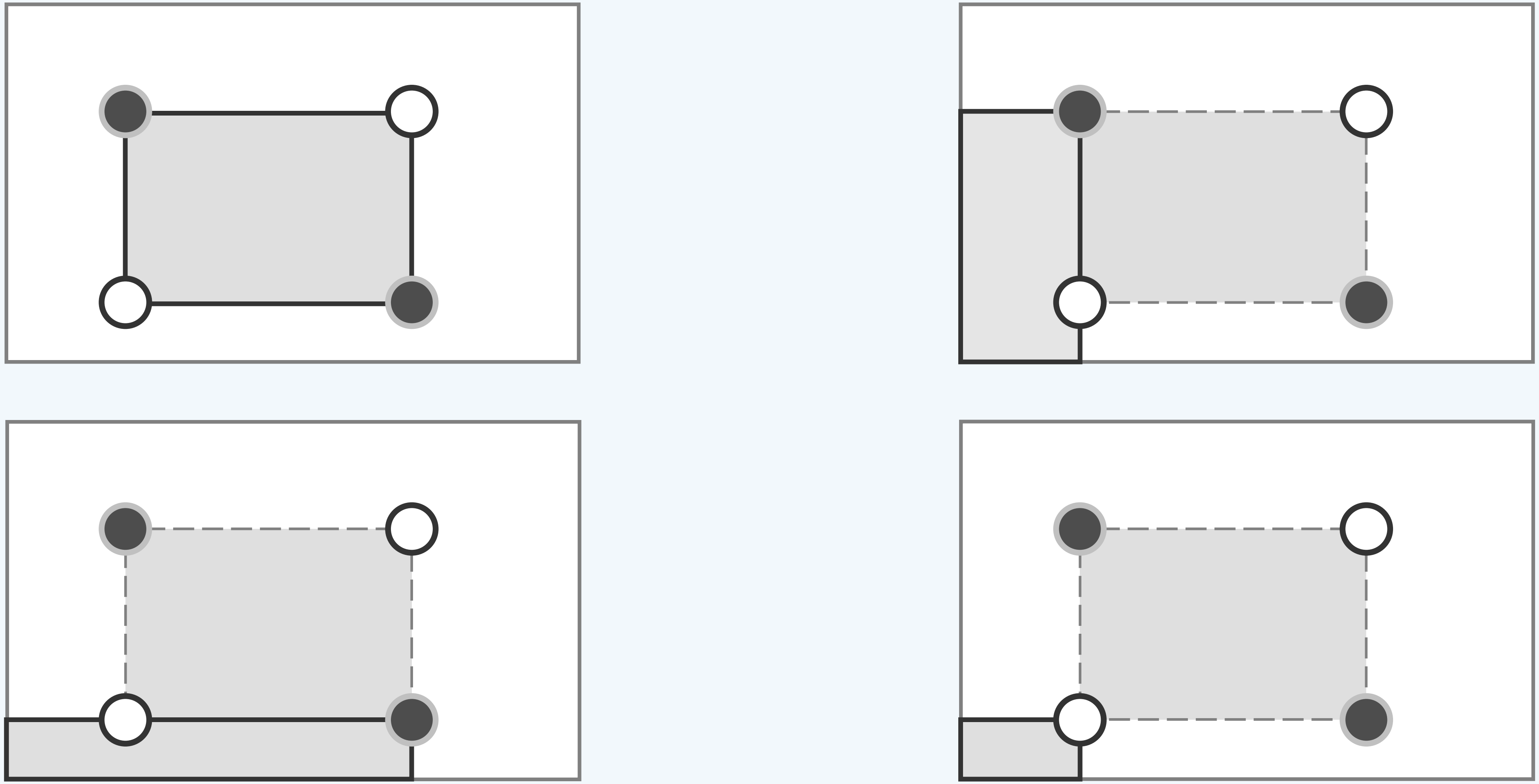
- 此时，计数法还能行得通吗？

- 否则，有无其它方法？



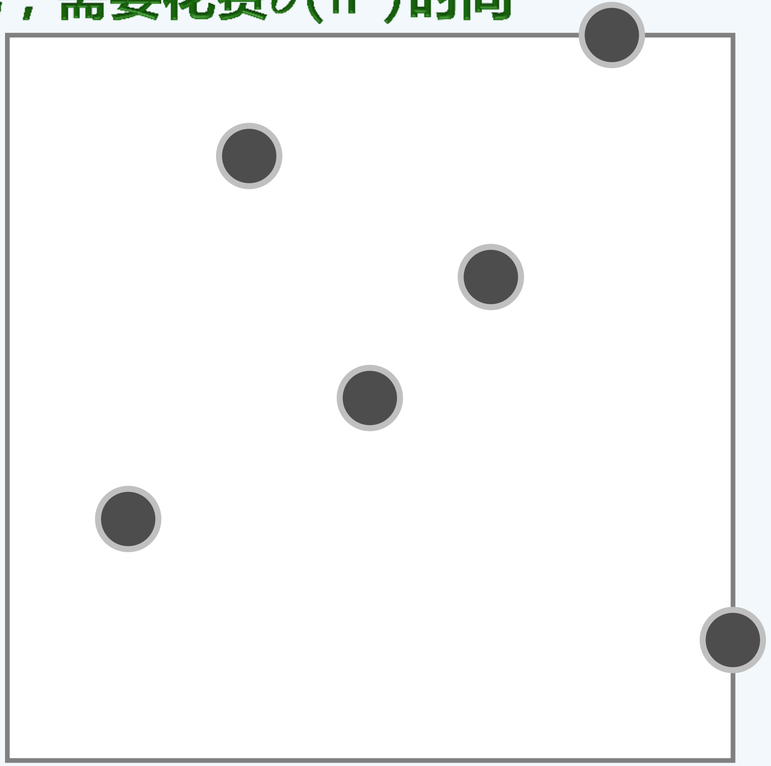
容斥原理

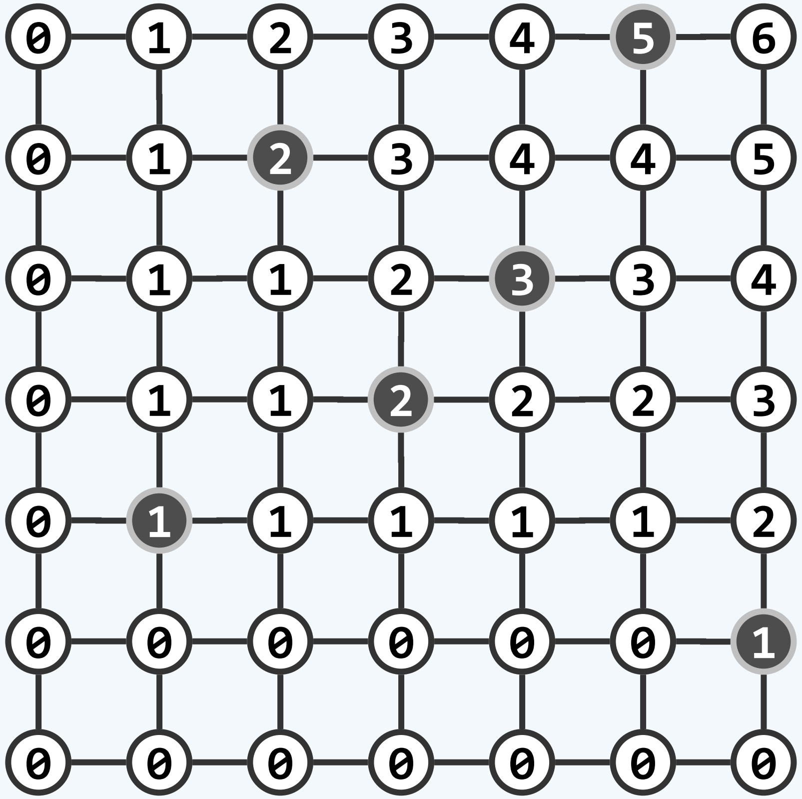
❖ 就原理而言，计数法不难推广至二维甚至更高维



覆盖

- ❖ 若u ≤ x且v ≤ y，则称点(u, v)被点(x, y)覆盖 dominated
- ❖ 预处理：对每个点(x, y)，记下 $n(x, y) = |\ (0, x] \times (0, y]\ \cap P\ |$
- ❖ 为此，需要花费$O(n^2)$时间



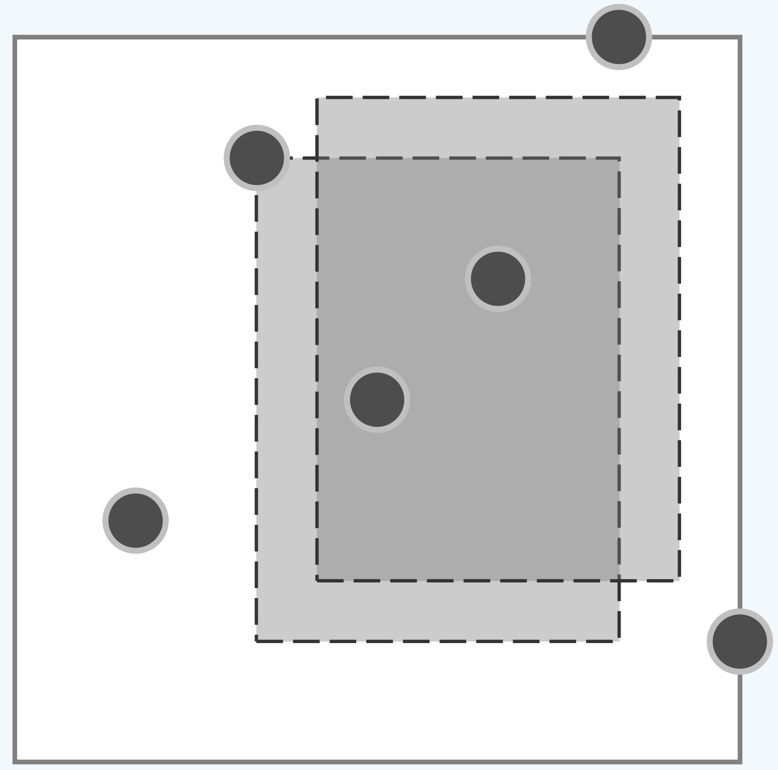


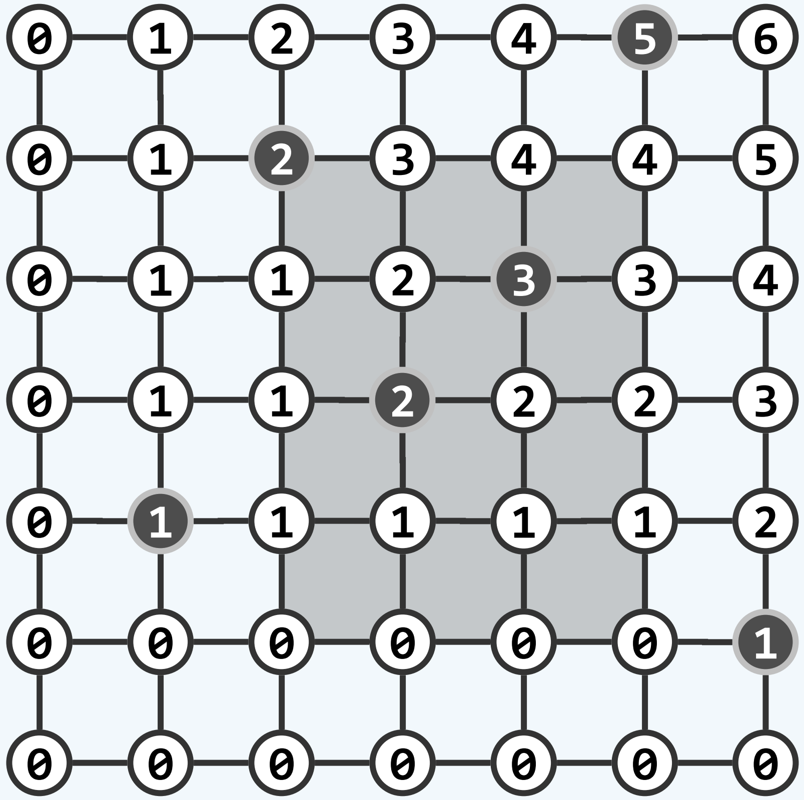
容斥原理

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

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

- 如何推广至全闭的矩形区域？若允许垂直或水平共线的点呢？

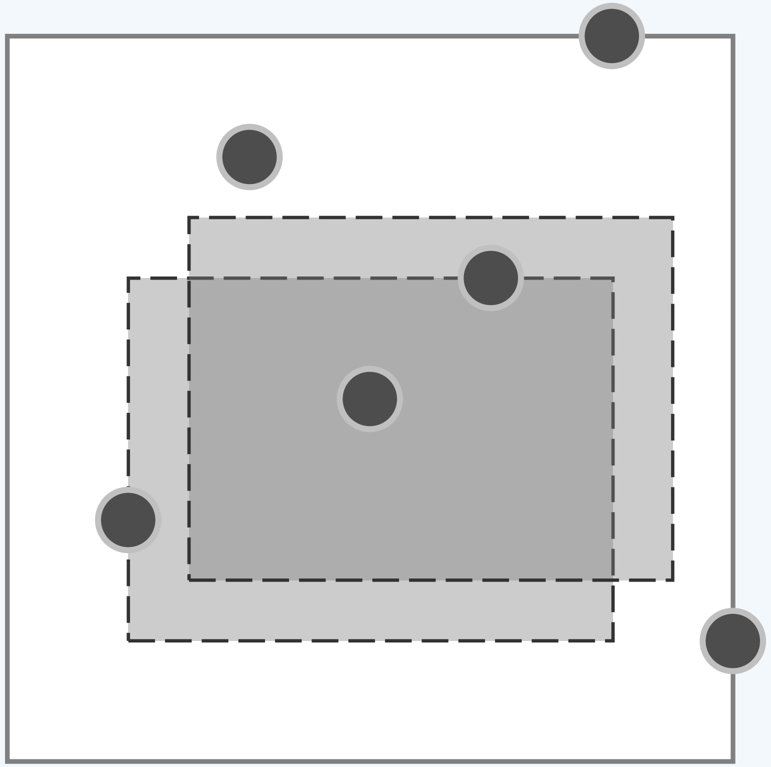


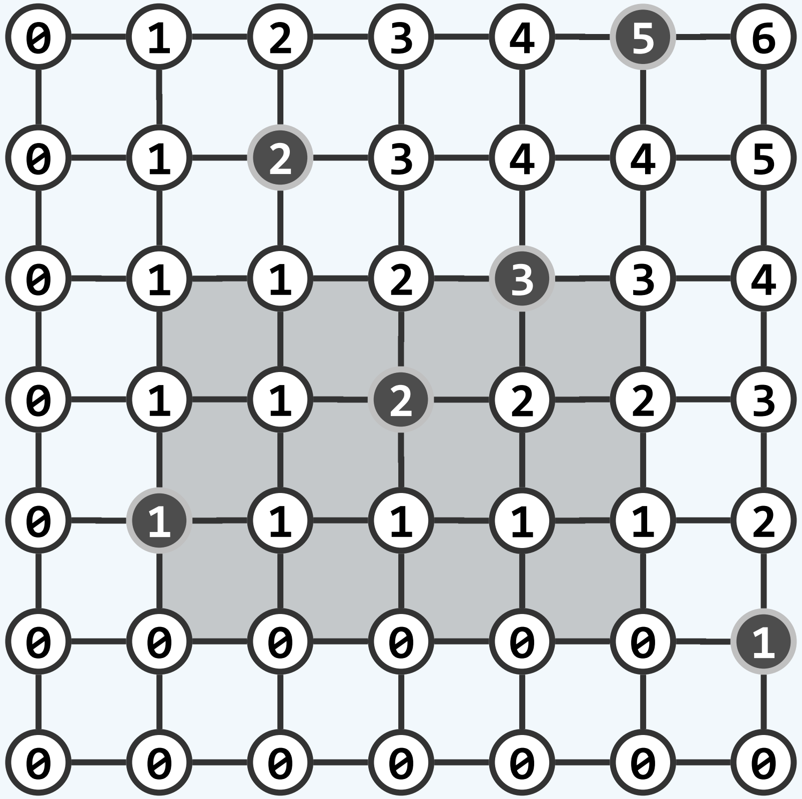


容斥原理

❖ 查询非常快，只需 $\boxed{\mathcal{O}(\log n)}$ 时间；但共需 $\boxed{\Theta(n^2)}$ 空间——倘若n较大，或者空间的维度更高...

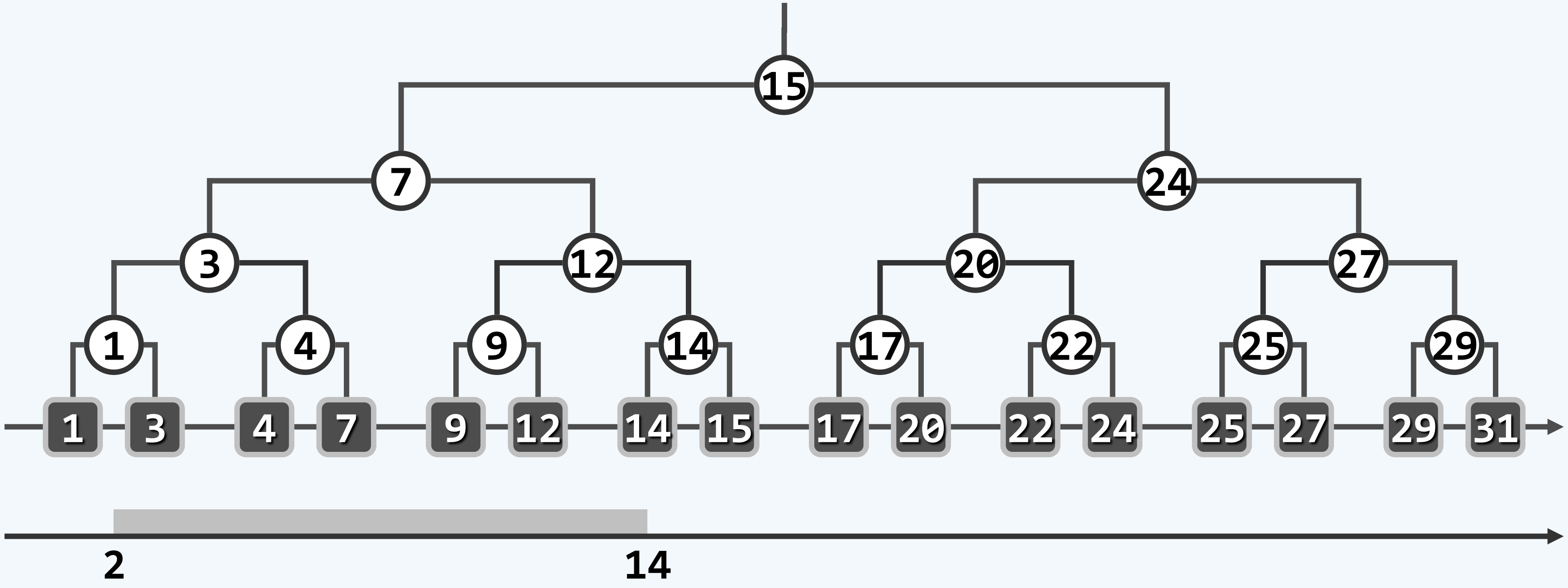
❖ 还是再次回到 $\boxed{\text{一维情况}}$，找出更为一般性的方法，以便推广...





BBST：结构

- 每个内部节点v，记录相应的划分位置x(v)
- 有序性：$\boxed{\boxed{L}Tree(v) \;\leq\; x(v) \;<\; \boxed{R}Tree(v)}$
- 比如x(v) = 12，有max{9, 12} ≤ 12 < min{14, 15}
- 以R = [2, 14]为例，范围查找如何实现？



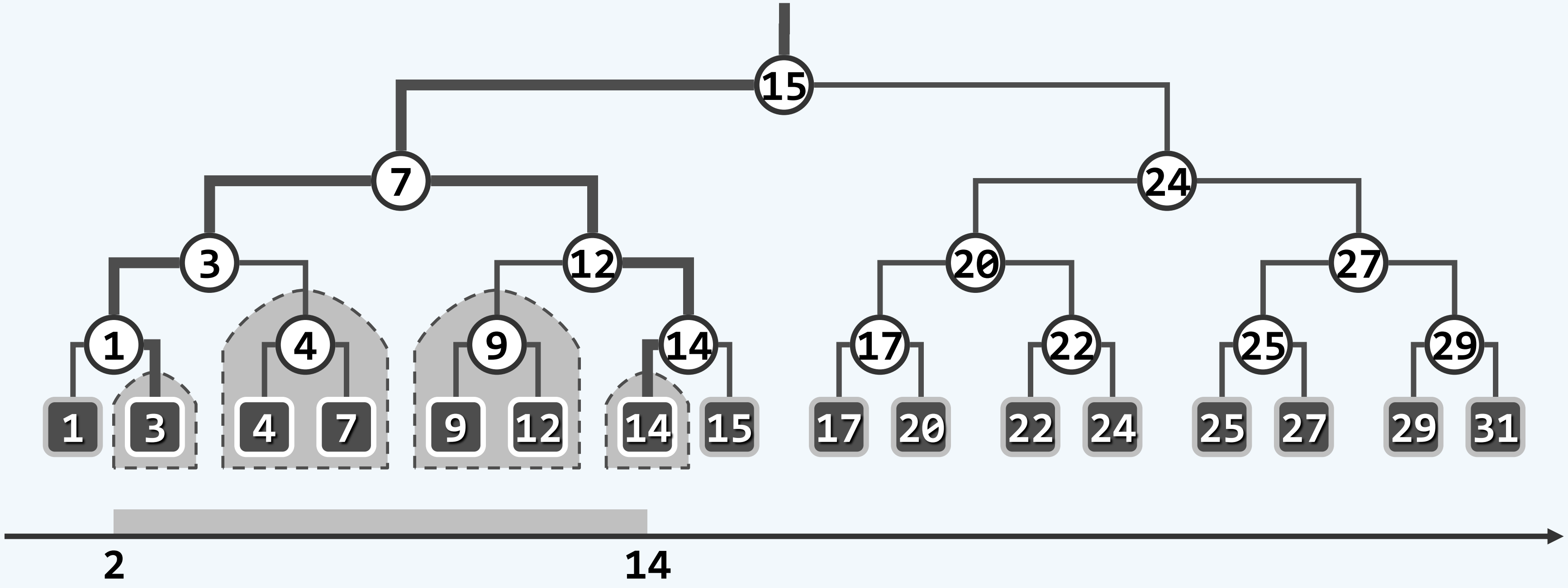
BBST：查找

❖ 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



BBST：效率

- 预处理 $\mathcal{O}(n\log n)$
- 空间 $\mathcal{O}(n)$
- 查询 $\boxed{\mathcal{O}(r + \log n)}$, r = 查找命中的点数 输出敏感