07-C2

搜索树应用

范围查询：二维

昔者明王必尽知天下良士之名；既知其名，又知其数；既知其数，又知其所在。

邓俊辉
deng@tsinghua.edu.cn

Planar Range Query

❖ **给定平面点集** $P = \{ p_1, p_2, p_3, \ldots, p_n \}$

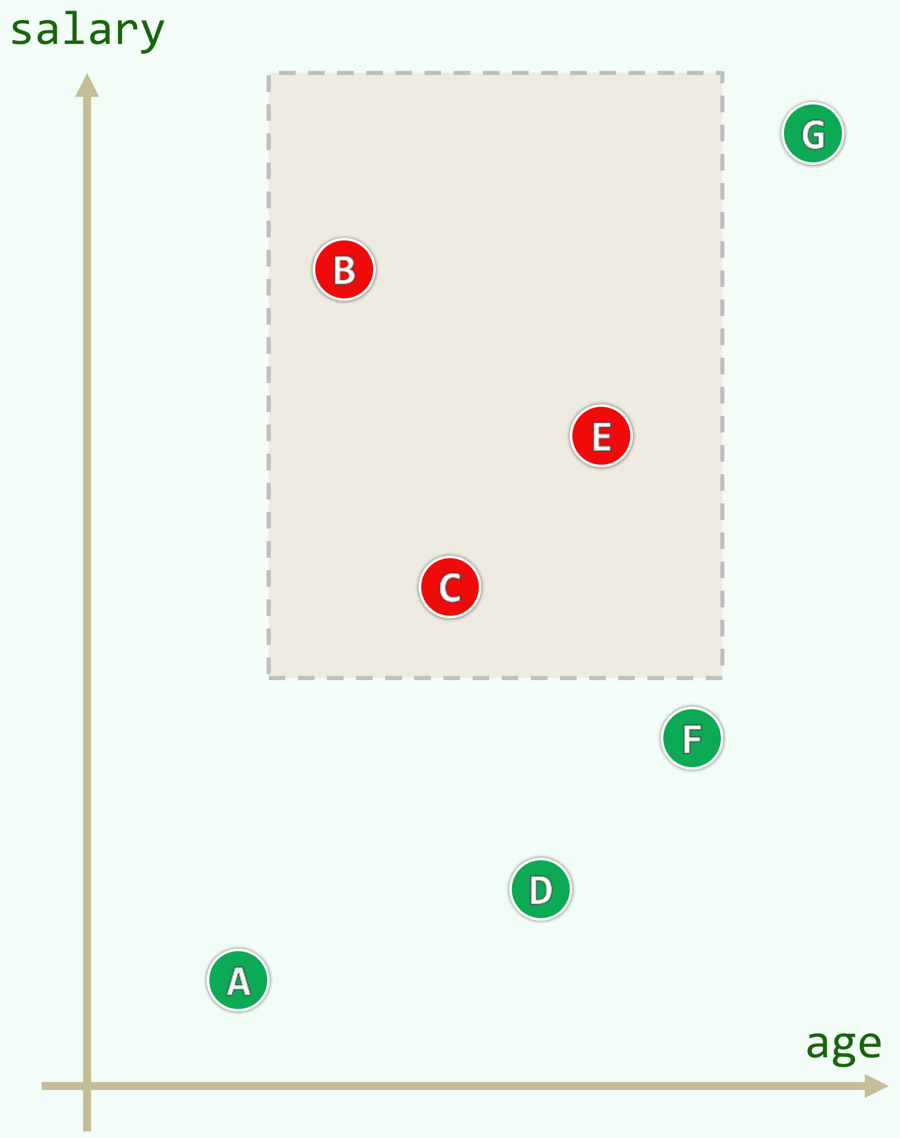
❖ **对任何矩形区域** $R = (x_1, x_2] \times (y_1, y_2]$

- **计数/COUNTING:** $|R \cap P| = ?$
- **报告/REPORTING:** $R \cap P = ?$

❖ **二分查找，对此问题无能为力**

❖ **如果只考虑计数查询**

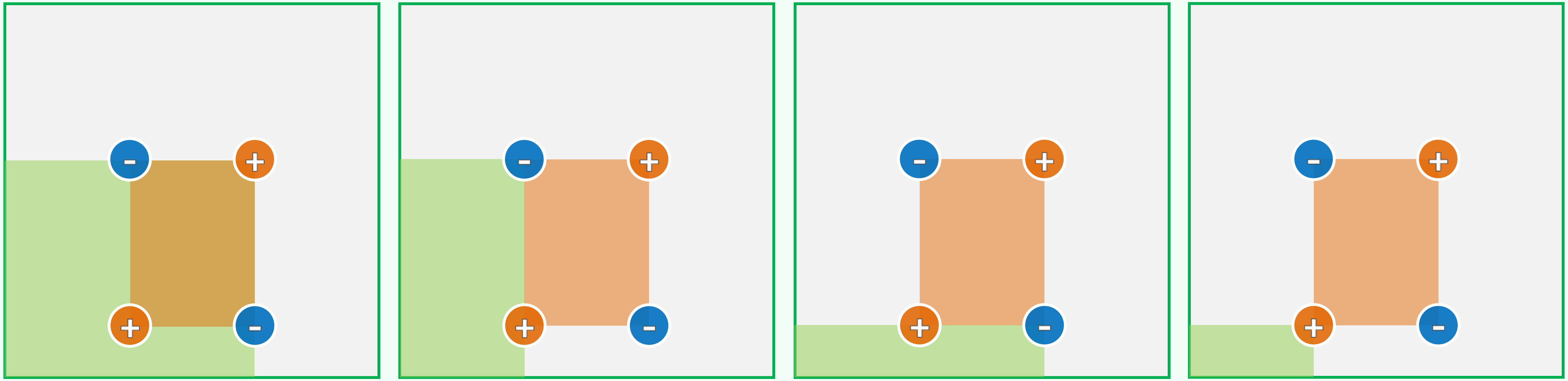
或许可以借助容斥原理...



Preprocessing

❖ 对任何一点 (x, y)，令 $n(x, y) = \left|((0, x] \times (0, y]) \cap P\right|$

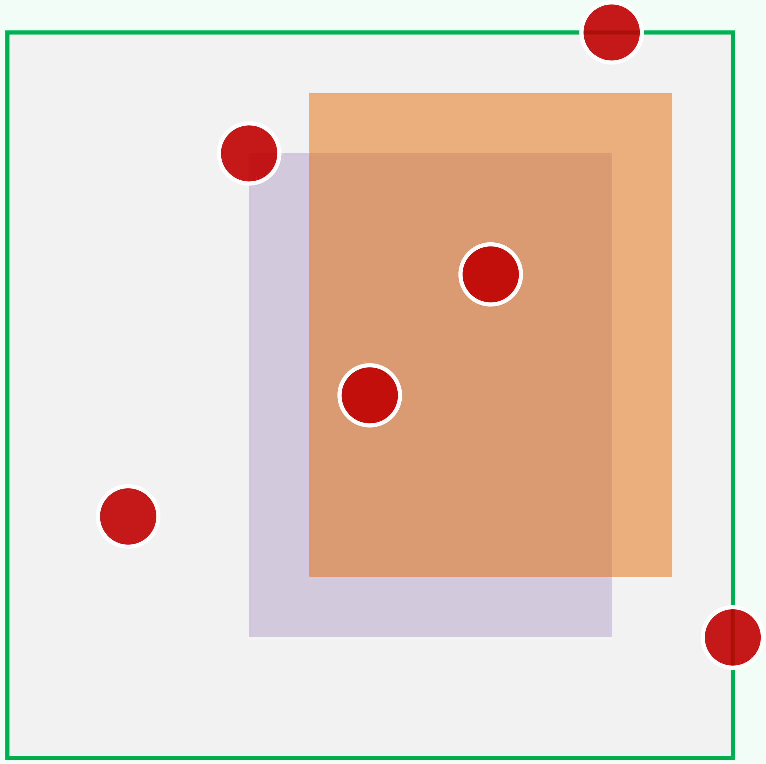
❖ 若共有 n 个输入点，则需预先计算并记录 $\Omega(n^2)$ 项，至少耗费 $\Omega(n^2)$ 时间和空间

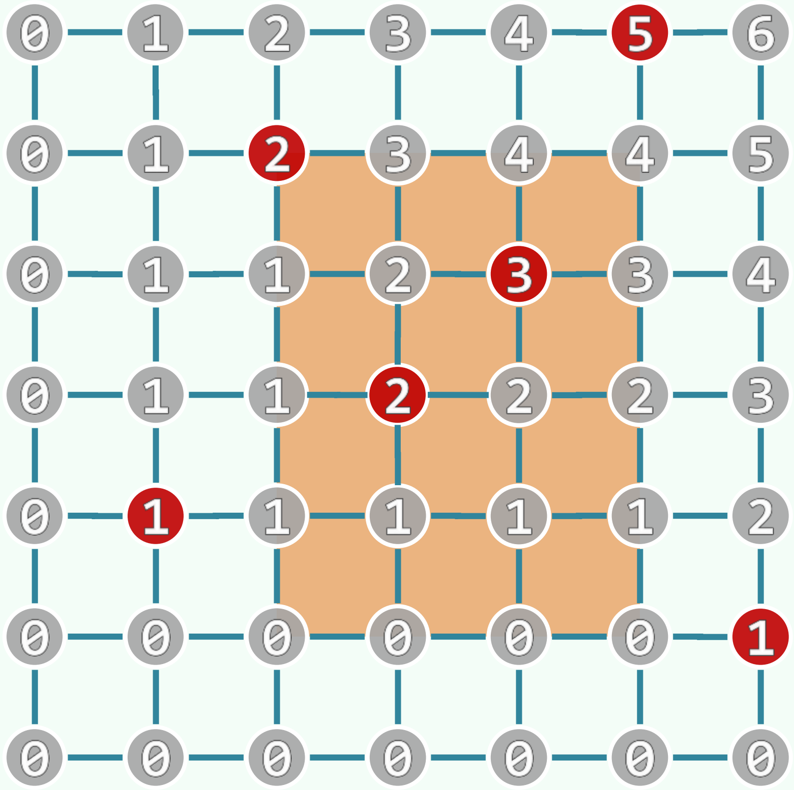


Domination

❖ 如果 $u \leq x$ 且 $v \leq y$，则称点 (u, v) 被点 (x, y) **覆盖**

❖ 通过二分查找，查询矩形的每个角点均可在 $\mathcal{O}(\log n)$ 时间内，**吸附**（Snap）到预记录过的某个点

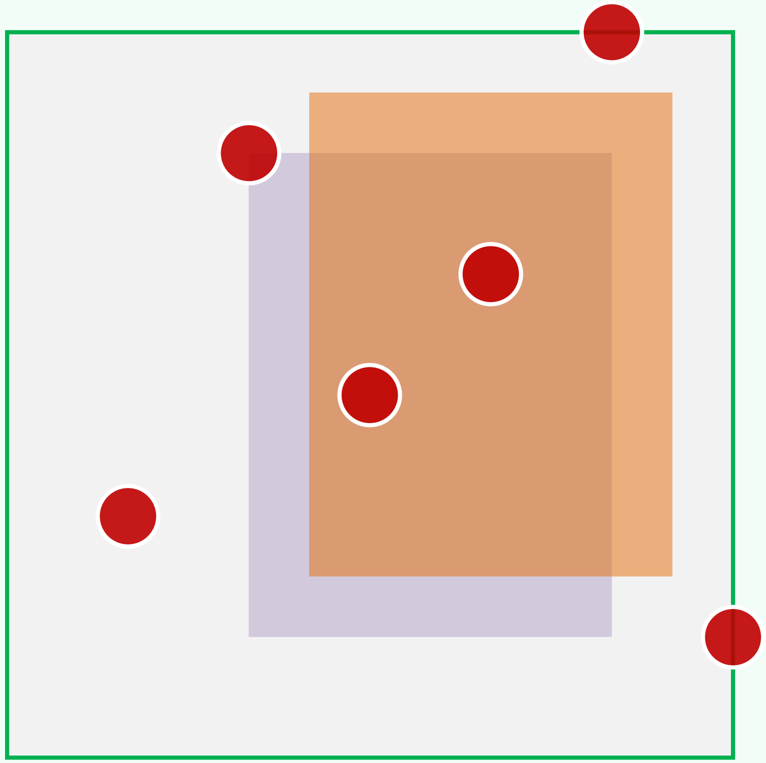


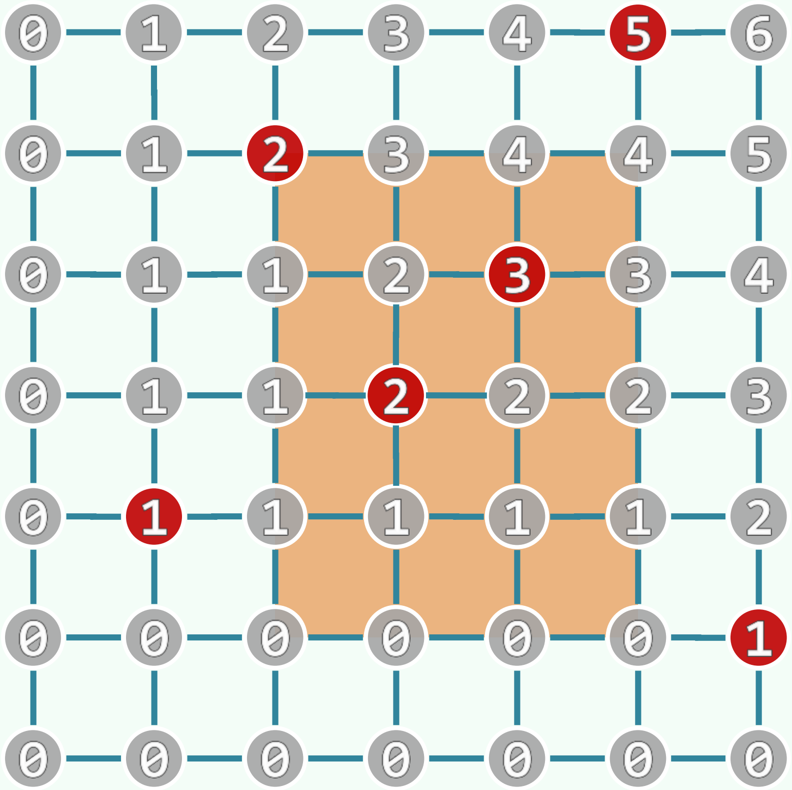


Inclusion-Exclusion Principle

❖ 于是，对任何的矩形查询范围 $\mathcal{R} = (x_1, x_2] \times (y_1, y_2]$，都有

$$\left|\mathcal{R} \cap \mathcal{P}\right| \;=\; n(x_1, y_1) + n(x_2, y_2) - n(x_1, y_2) - n(x_2, y_1)$$





Performance

❖ 如此，每次查询只需 $\mathcal{O}(\log n)$ 时间；为此，需要耗费 $\Omega(n^2)$ **空间**（并随维度呈**指数**增长）

❖ 为找到可行的方法，我们还是需要重新审视**一维**的情况...