07-F3

搜索树应用

kd-树：复杂度

肉眼看不清细节，但他们都知道那是木星所在的位置，
这颗太阳系最大的行星已经坠落到二维平面上了

有人嘲笑这种体系说：为了能发现这个比例中项并组成政府
共同体，按照我的办法，只消求出人口数字的平方根就行了

邓俊辉
deng@tsinghua.edu.cn

预处理 + 空间消耗

❖ 时间 $T(n) = 2*T(n/2) + \mathcal{O}(n)$

$= \mathcal{O}(n\log n)$

❖ 树高 $= \mathcal{O}(\log n)$

❖ 空间 $= 1$

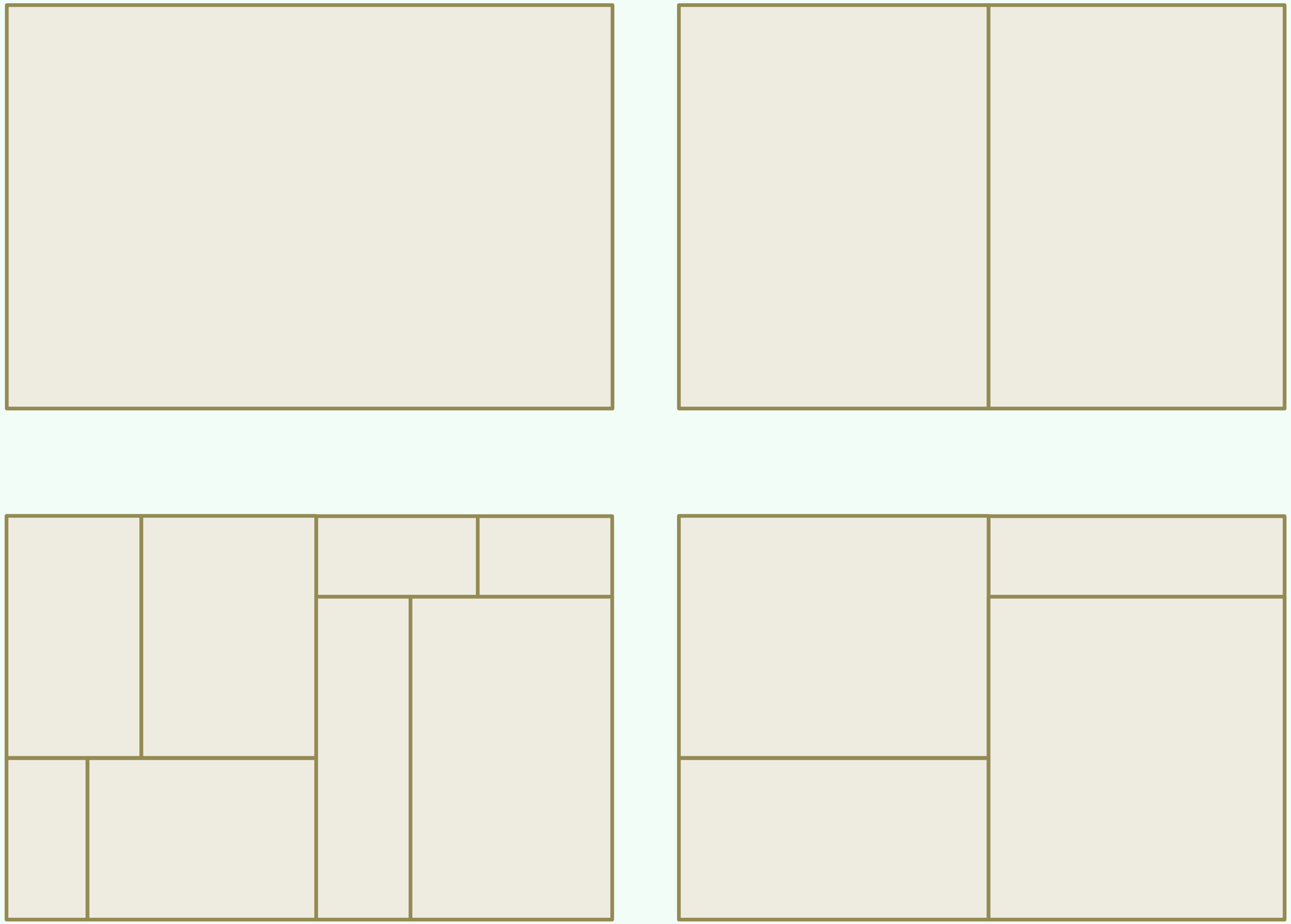
$+ \quad 2$

$+ \quad 4$

$+ \quad \ldots$

$+ \quad \mathcal{O}(2^{\log n})$

$= \quad \mathcal{O}(n)$



查询时间（1/2）

❖ 声明：查询 = 查找+报告 = $\mathcal{O}(\sqrt{n}+r)$

❖ 以下均就**最坏情况**而言...

❖ 查找的时间成本，决定于

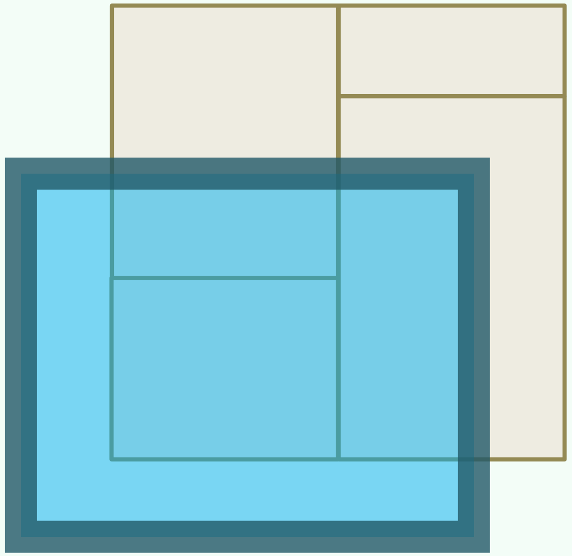
$Q(n)$ = **递归**调用的总次数，亦即

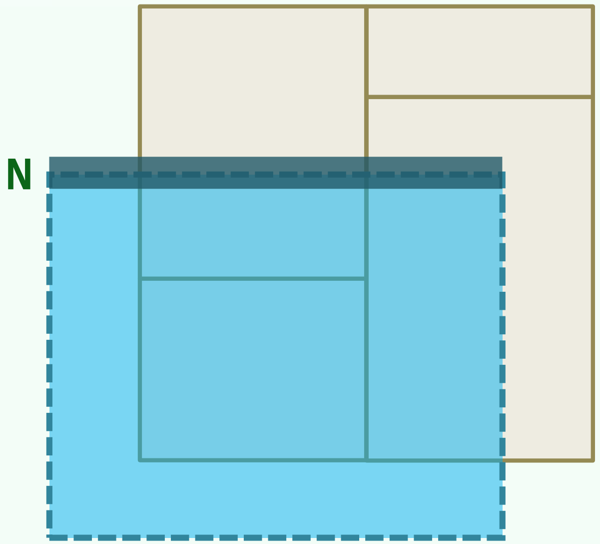
与R**边界相交**的子区域（节点）总数

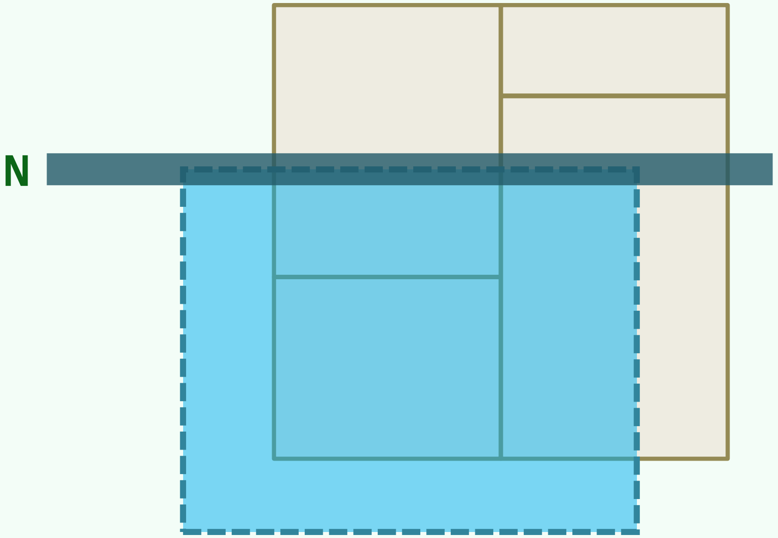
❖ 观察：**任何一段**边界所引发的递归总数，与 $Q(n)$ 渐近相等

因此，只需考查**其中一段**，WOLG，北方的那段**N**

❖ 进一步地，可将**N**放大为一条**直线**







查询时间（2/2）

❖ 以下，我们来导出 $Q(n)$ 的递推式

❖ 很遗憾，通常“由父及子”的思路

在这里行不通，需改成“由祖及子”...

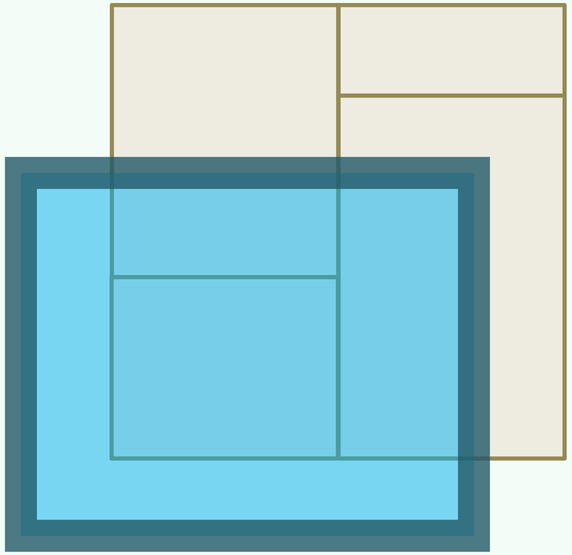
❖ 幸运的是

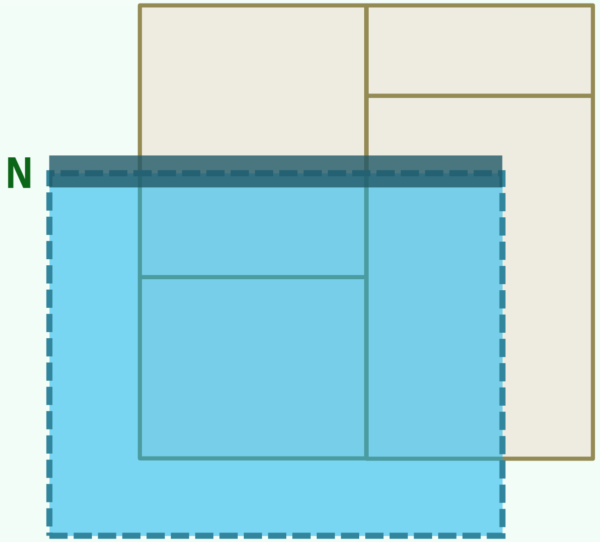
奇数层上发生递归的总次数，也与 $Q(n)$ 渐近相等

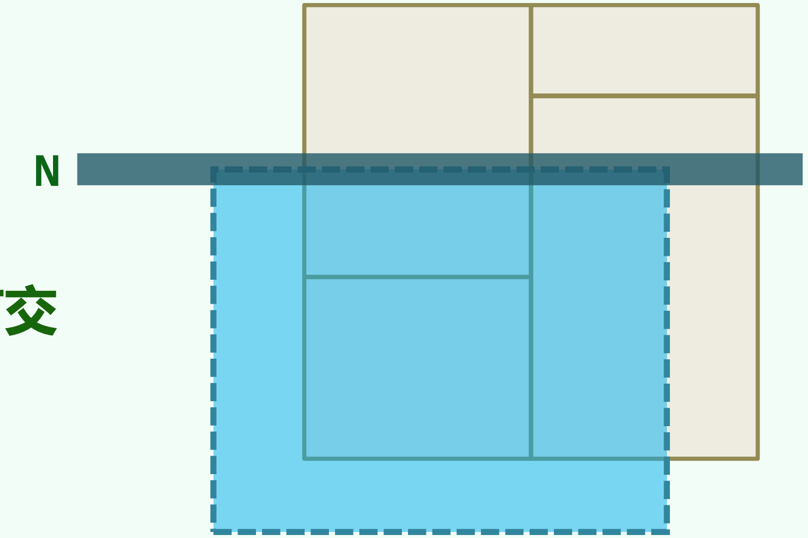
❖ 观察：

任一区域若与N有交，则在其四个孙子中，至多会有两个也与N有交

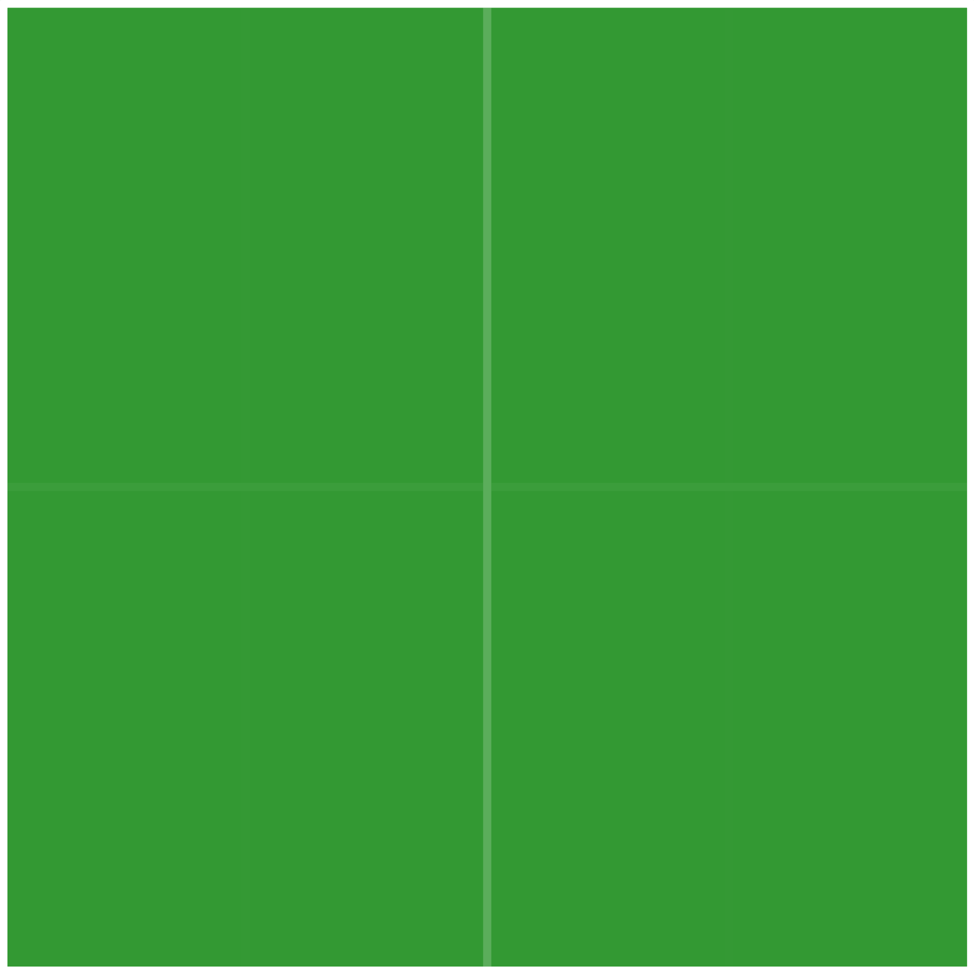
❖ 亦即：$Q(n) \;=\; 2 \cdot Q(n/4) + \mathcal{O}(1) \;=\; \mathcal{O}(\sqrt{n})$

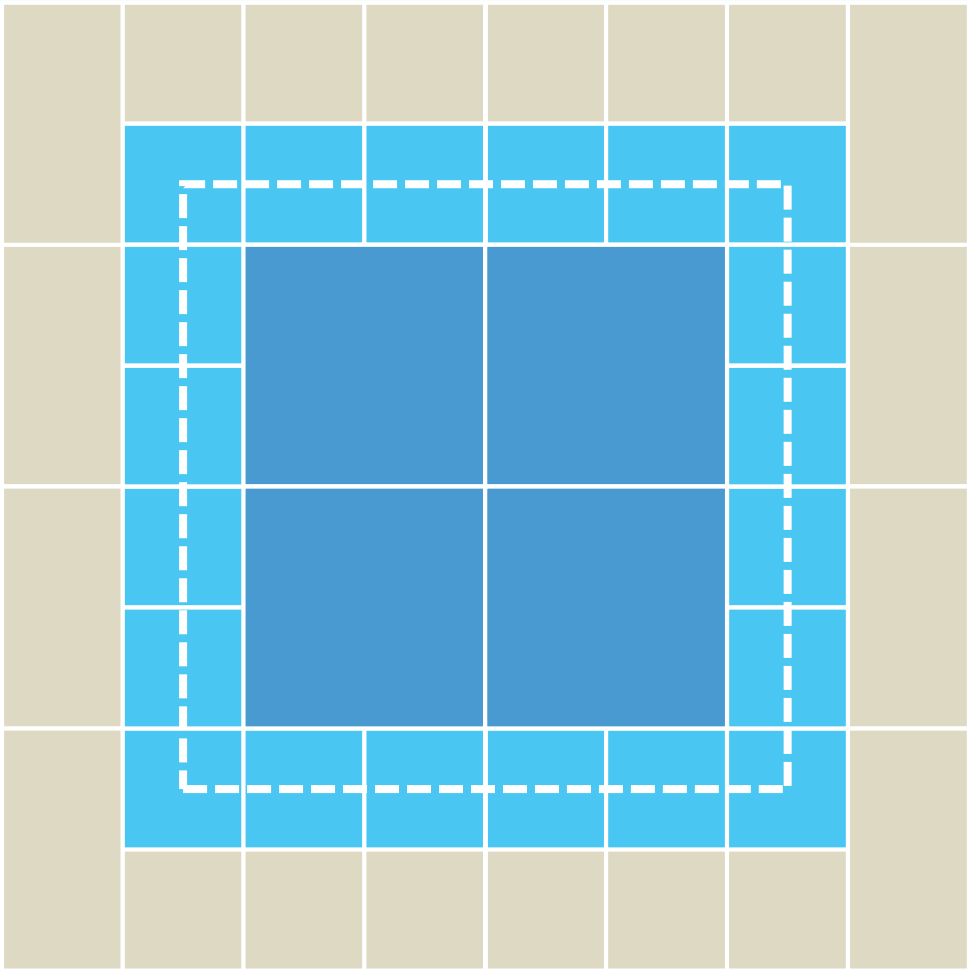






最坏情况





更高维度

❖ 2d-树可否推广，以支持高维空间的范围查询？

在这类场合，其时间、空间效率如何？

❖ 实际上，推广并不难：

在k维空间中，只需循环地依次沿第1、第2、第3、...、第k维切分

❖ 在d维欧氏空间 $\mathcal{E}^d$ 中，对于任意给定的 n 个点，我们都可以

- 在 $\mathcal{O}(n\log n)$ 时间内构造出
- 一棵占用 $\mathcal{O}(n)$ 空间的kd-树，并借助它
- 在 $\mathcal{O}(r + n^{1-1/d})$ 时间内完成每次正交范围查询