07-F1

搜索树应用

kd-树：二维

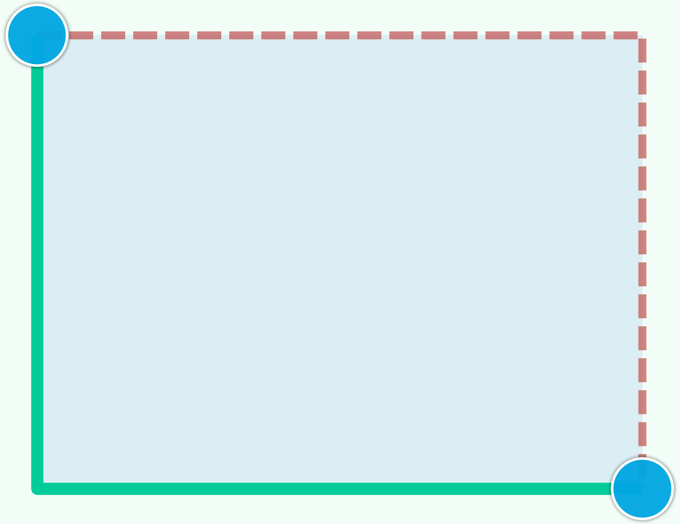
韦小宝跟著她走到桌边，只见桌上大白布上钉满了几千枚绣花针，几千块碎片已拼成一幅完整无缺的大地图，难得的是几千片碎皮拼在一起，既没多出一片，也没少了一片

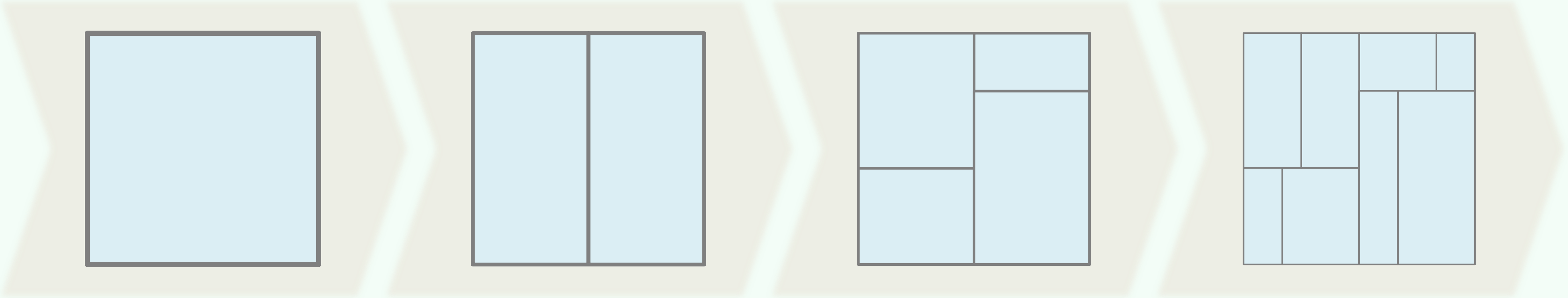
夫仁政，必自經界始...經界既正，分田制祿可坐而定也

邓俊辉
deng@tsinghua.edu.cn

思路

- 为利用BBST来支持二维的区域查询，可以
  递归地**划分**平面，并将分出来的**矩形区域**组织为一棵**二叉树**
- 首先，**根节点**对应于整个平面
  然后，**交替地**按x、y坐标划分，直至...
- 为避免歧义，可约定每个矩形区域都是**左闭右开**、**下闭上开**

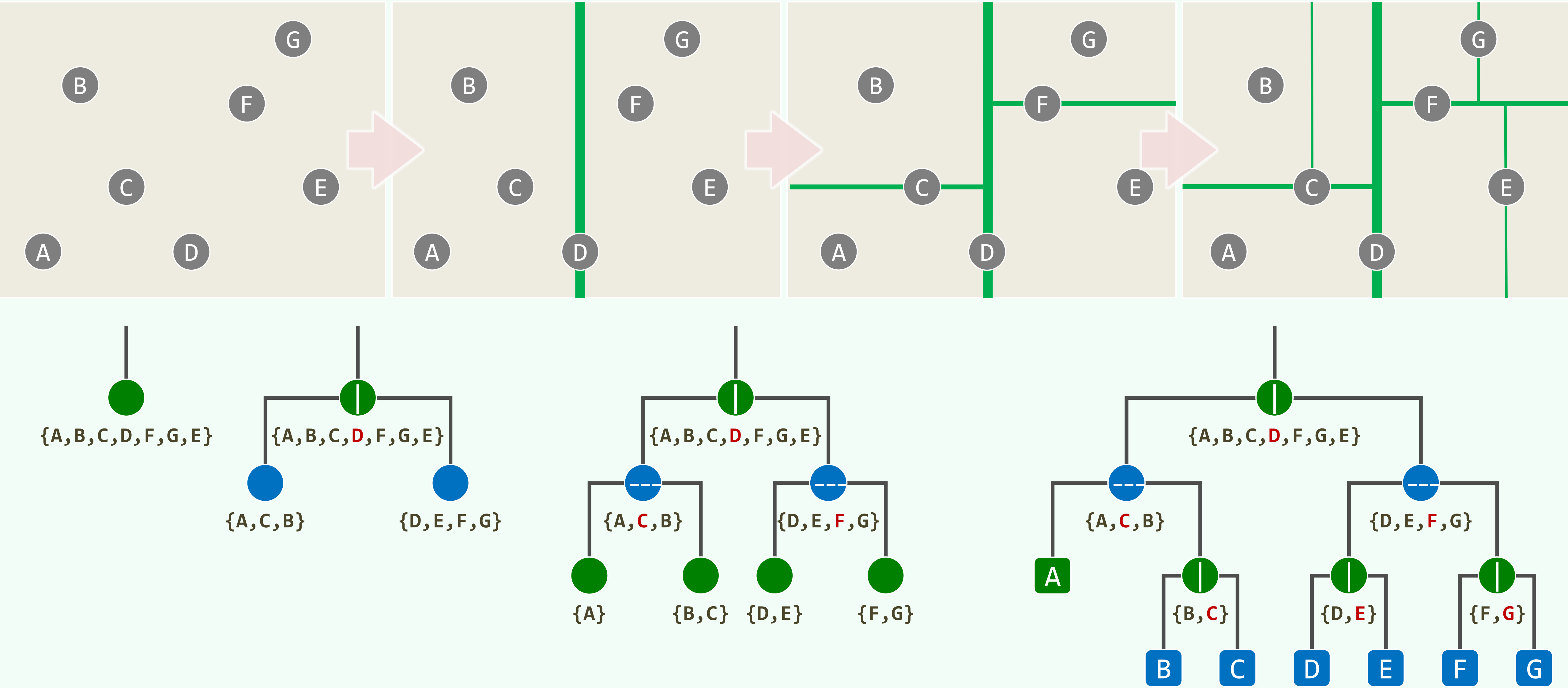




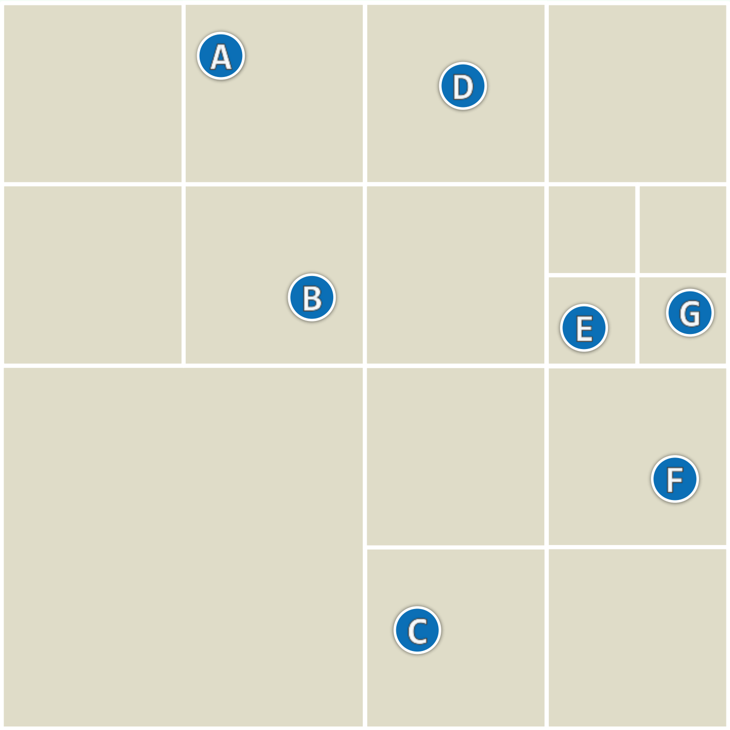
构造：算法buildKdTree(P,d)

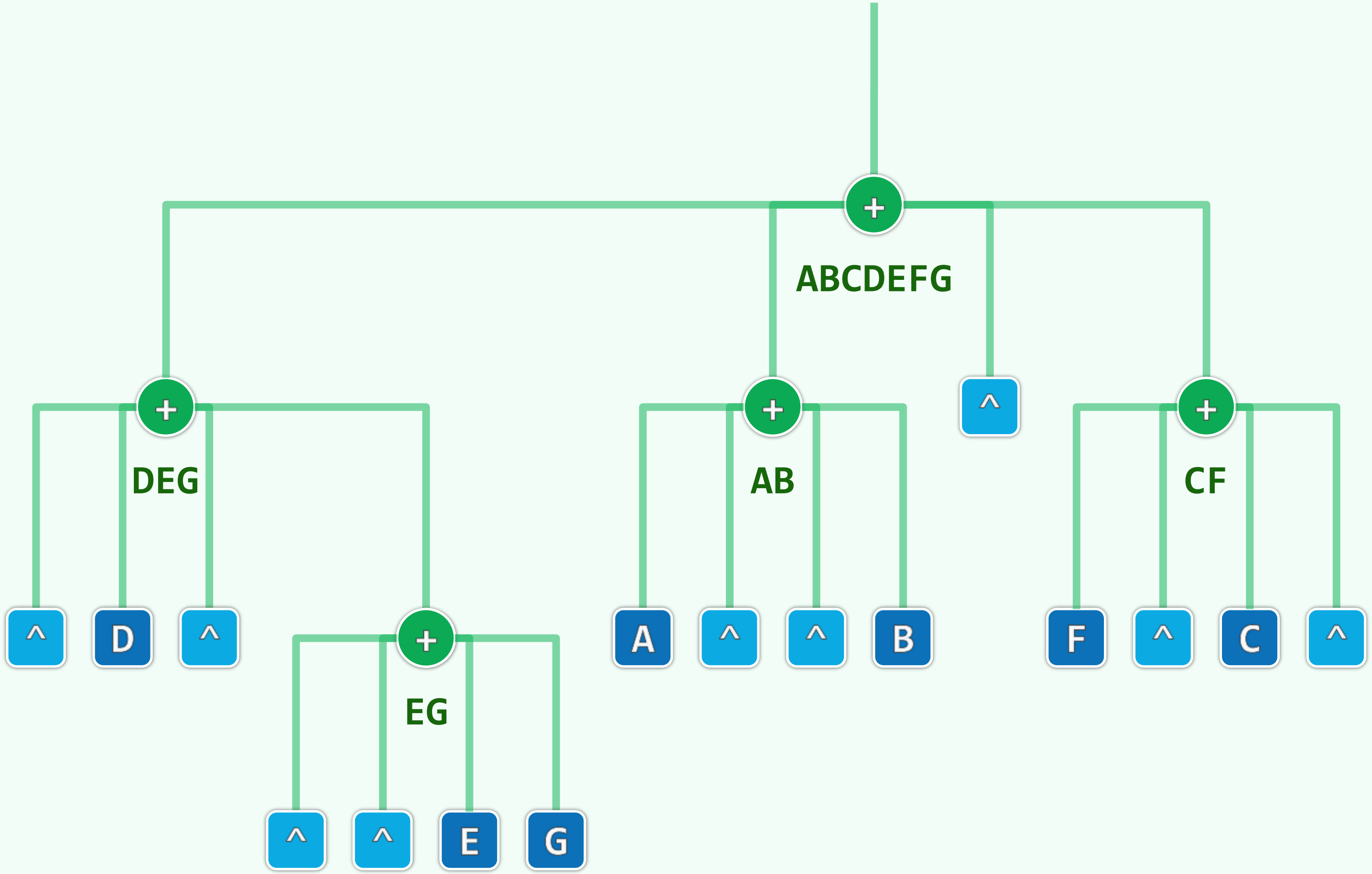
```
// Construct a 2d-tree for point set P at depth d
   if ( P == {p} ) return createLeaf( p )  //base
   Root = createKdNode()
   Root->SplitDirection = even(d) ? VERTICAL : HORIZONTAL
   Root->SplitLine = findMedian( root->SplitDirection, P )  //𝒪(n)!
   ( P1, P2 ) = divide( P, Root->SplitDirection, Root->SplitLine )  //DAC
   Root->LC  = buildKdTree( P1, d + 1 )  //recurse
   Root->RC  = buildKdTree( P2, d + 1 )  //recurse
   return( Root )
```

构造：实例



特例：Quadtree





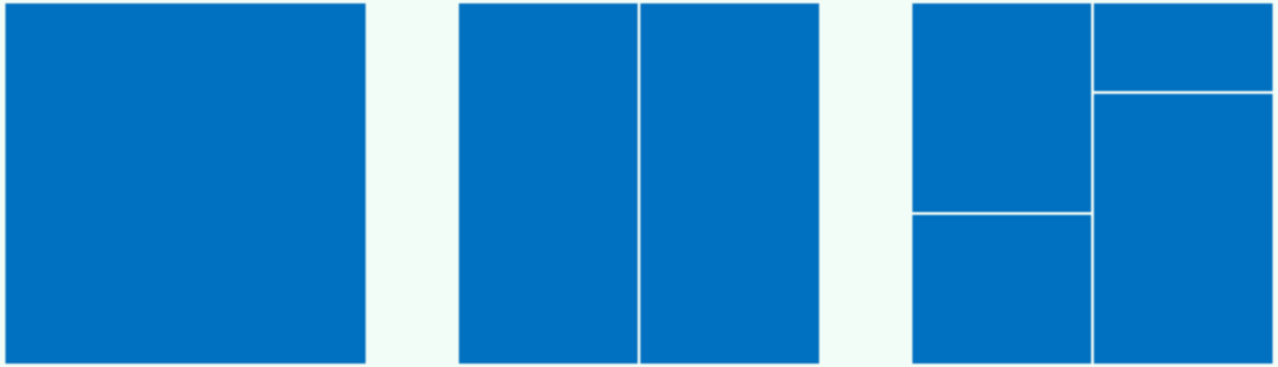
性质

- 树中的每个**节点** v，都对应于平面上的一个**矩形区域** $region(v)$ 以及子集 $P(v) = P \cap region(v)$
- 树高 $height(T) = \mathcal{O}(\log n)$
- 任何**同层**的节点 v 和 u，都有：

$$region(v) \cap region(u) = \varnothing$$

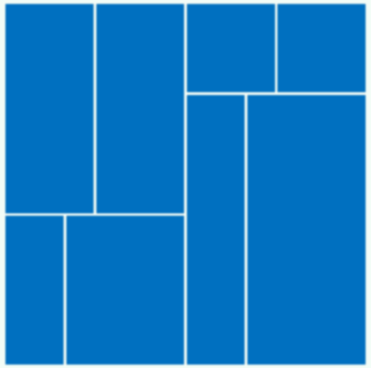
- 若**兄弟**节点 v 和 u 的父亲为 p，则有：

$$region(p) = region(v) \mathbin{\dot{\cup}} region(u)$$



- 同层的**所有**节点所对应的区域，无缝地**覆盖**了整个平面：

$$\forall\, 0 \le d \le height(T),\ \dot{\bigcup_{depth(v)=d}} region(v) = \mathcal{R}^2 \text{ 且 } \dot{\bigcup_{depth(v)=d}} P(v) = P$$



- 任何一次矩形查询的**解**，都可以表示为**若干个**节点所对应 $P(v)$ 的**并** //多少个?