8.高级搜索树

（xb2）kd-树：二维

邓俊辉

deng@tsinghua.edu.cn

构思

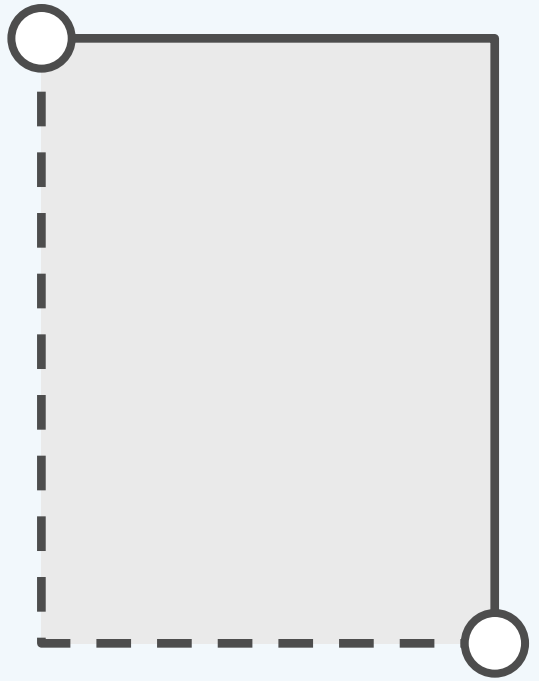
- 上述数据结构，如何扩展到二维？
- 通过递归的平面划分，构造kd-树！

  偶/奇数深度层：做垂直/水平划分

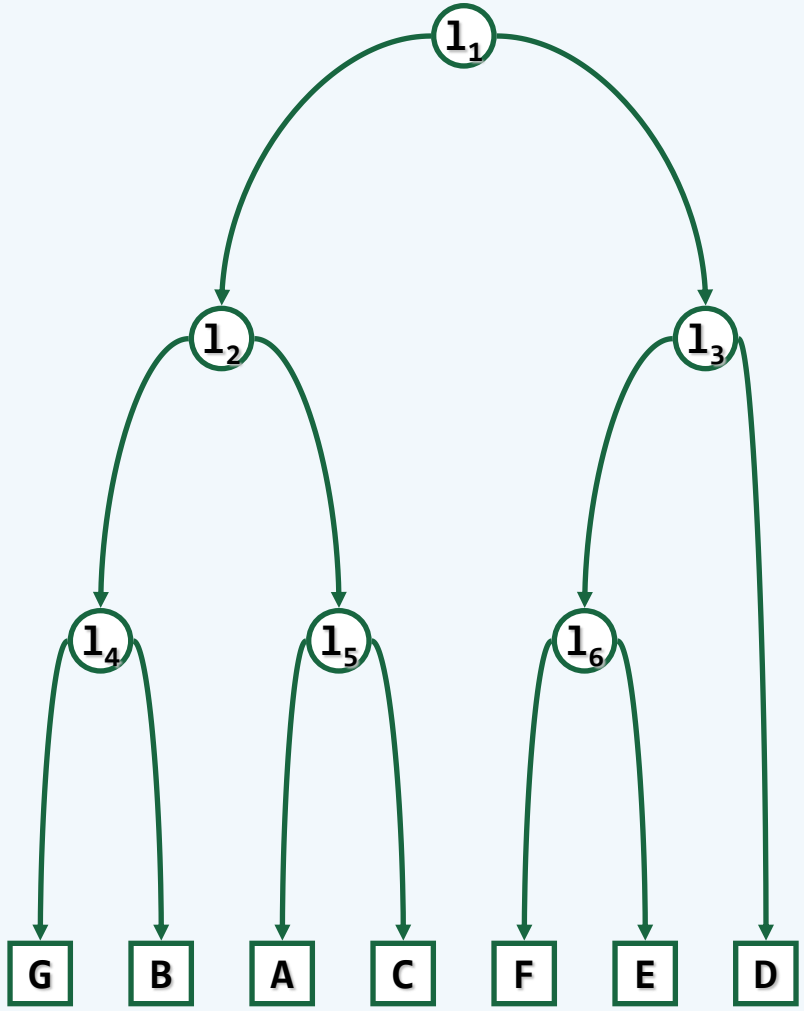
  两个子集规模尽可能接近median

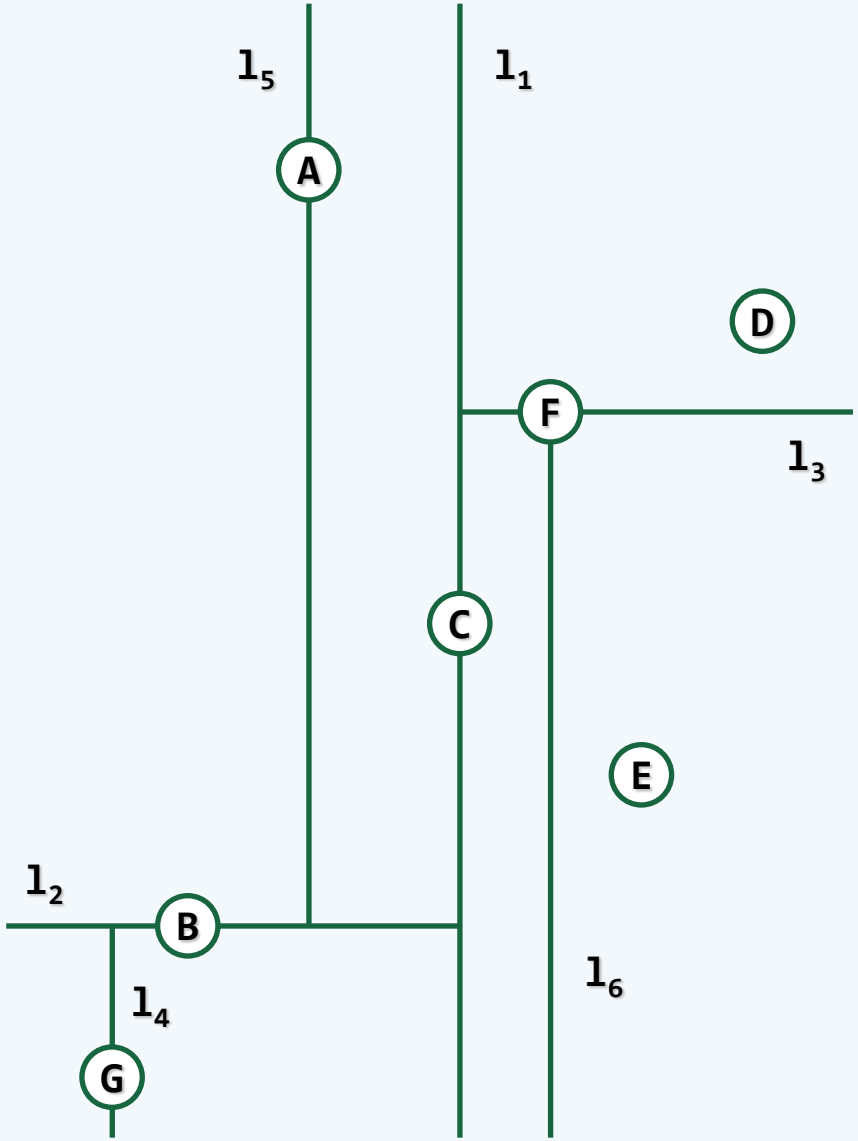
  半平面：左开右闭、下开上闭

  非退化约定：各点x坐标互异、y坐标互异



构思实例

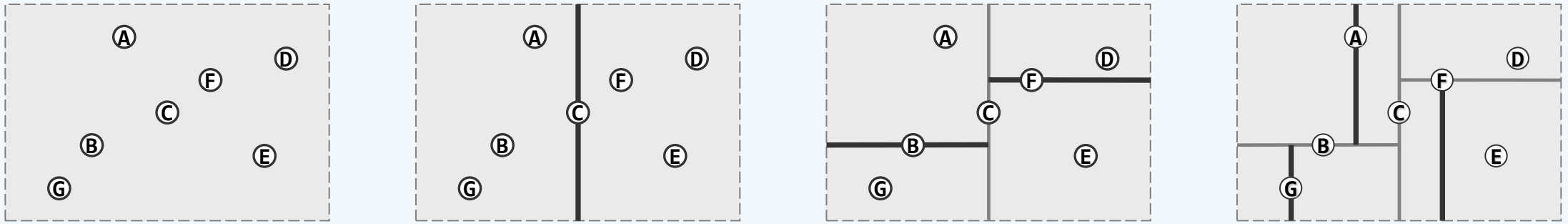


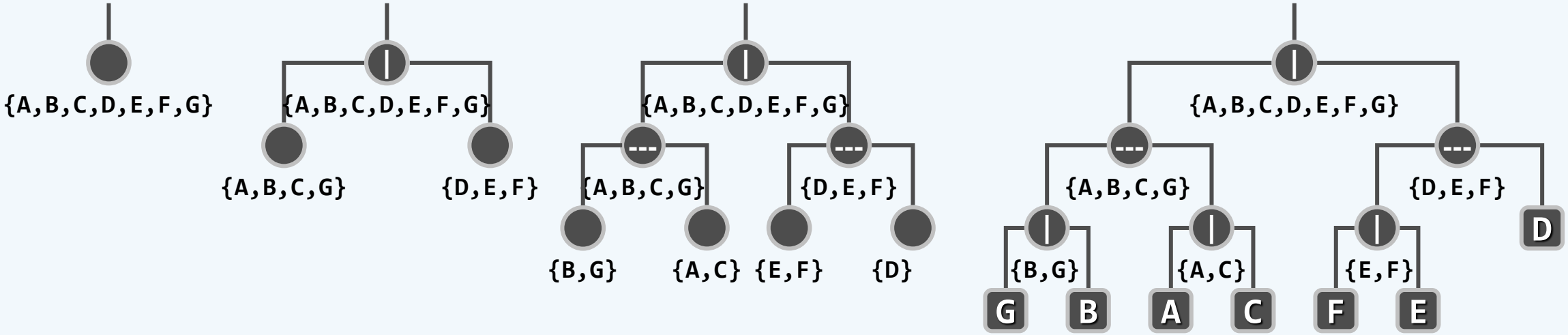


构造算法

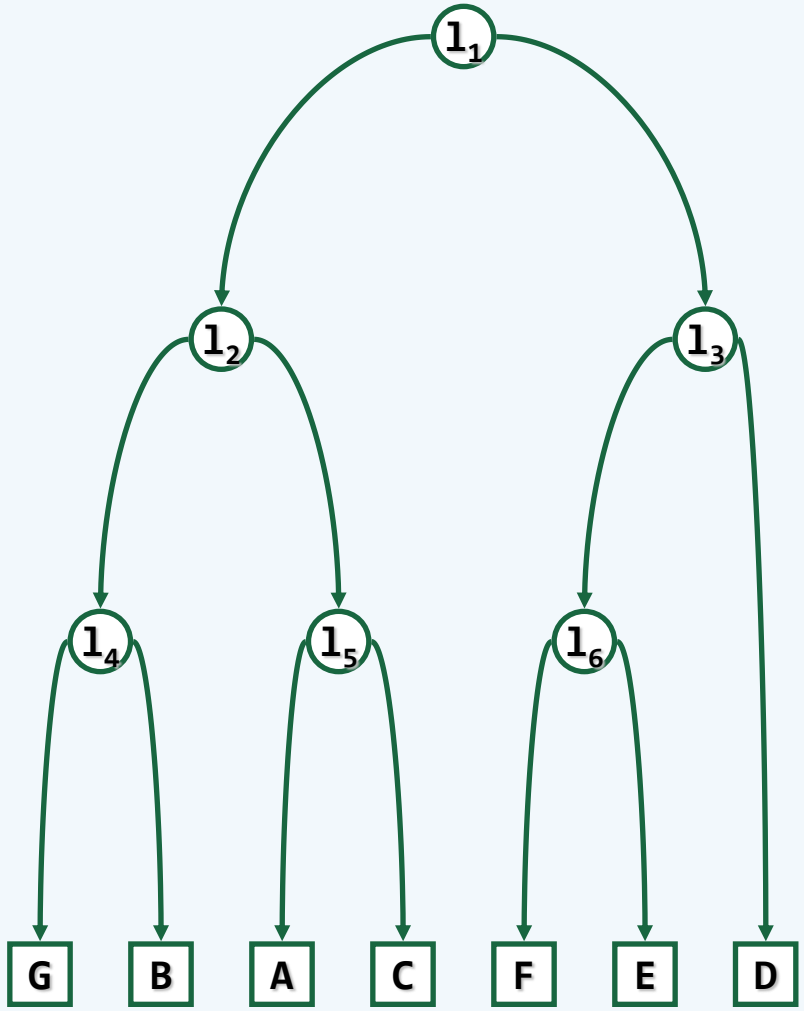
```
KdTree* createKdTee( P, depth ) { //自顶而下，递归地逐层构造
   if ( P == {p} ) return createLeaf( p ); //递归基：区域缩小至仅含一个点
   root = createKdNode(); //创建节点
   root->splitDirection = depth % 2 ? HORIZONTAL : VERTICAL; //切分方向
   root->splitLine = median( root->splitDirection, P ); //切分位置
   ( P1, P2 ) = divide( P, root->splitDirection, root->splitLine ); //切分
   root->lc = createKdTee( P1, depth + 1 ); //递归构造左或上子树
   root->rc = createKdTee( P2, depth + 1 ); //递归构造右或下子树
   return root;
}
```

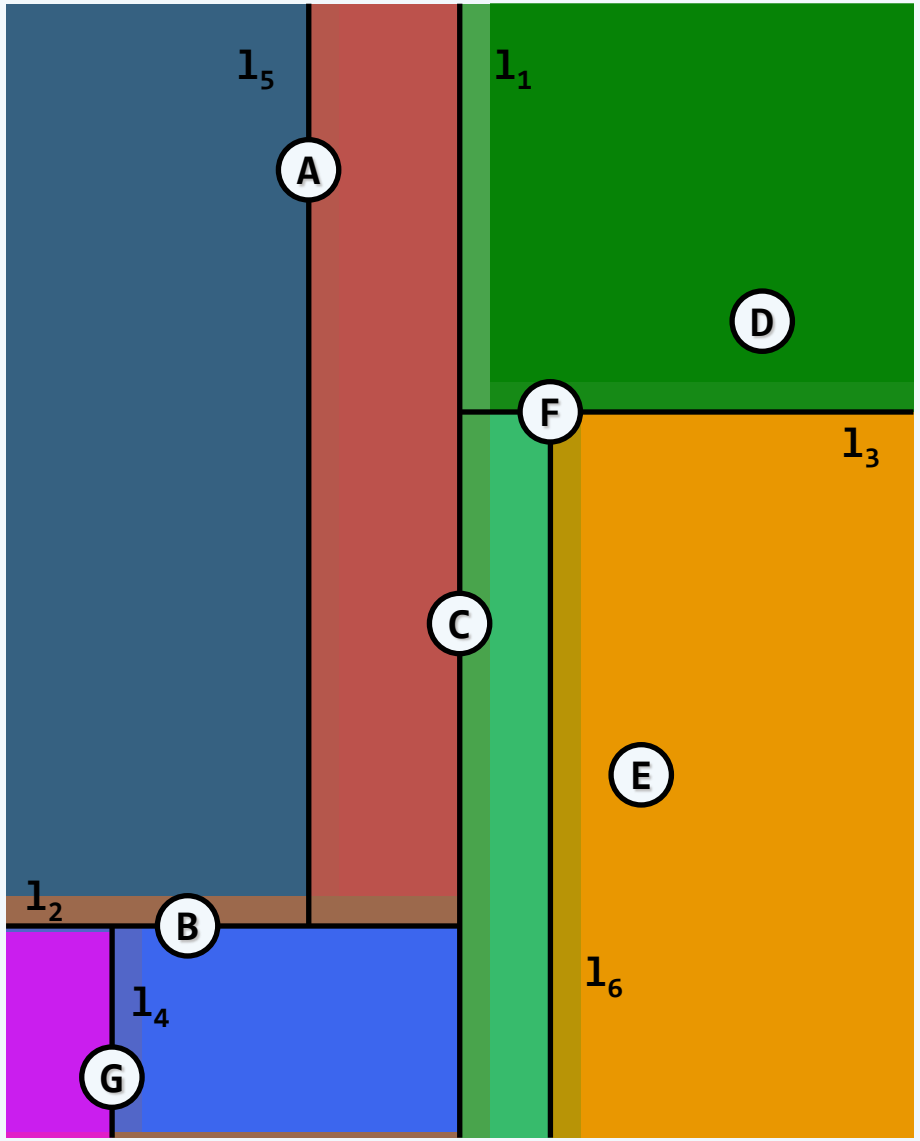
构造实例





构造实例



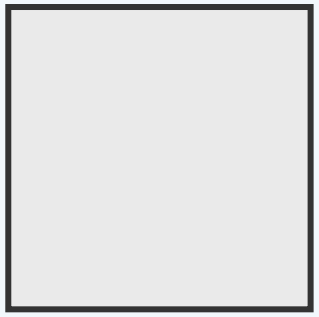
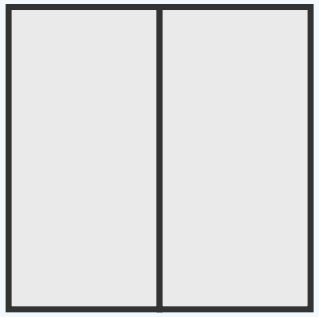
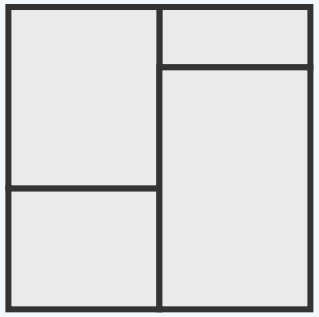
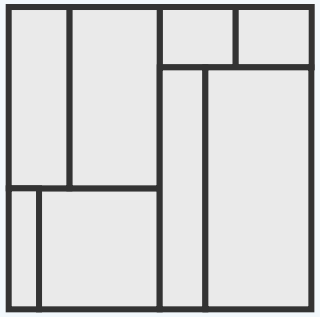


正则子集

- 树节点 ~ 矩形子区域
- 同一节点左、右孩子所对应子区域之并，即该节点对应的子区域
- 同一层的所有节点对应的子区域

  互不相交，而且

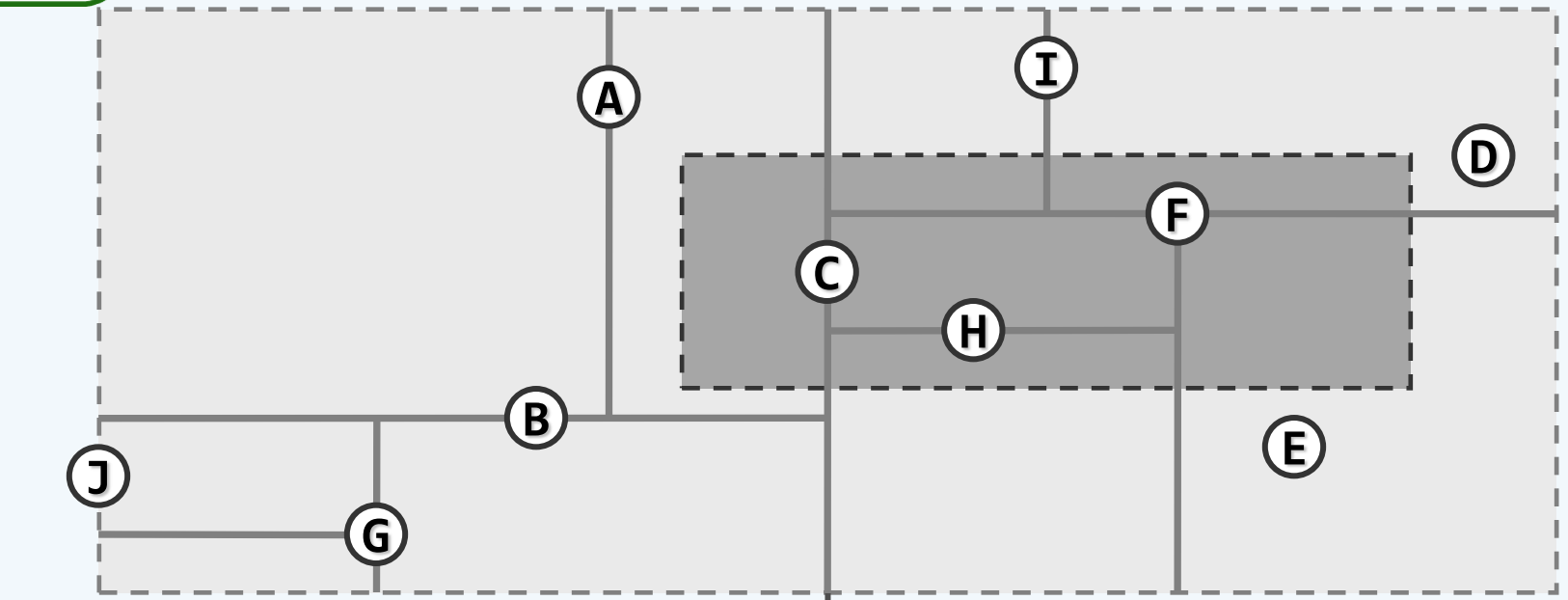
  其并恰好覆盖整个平面

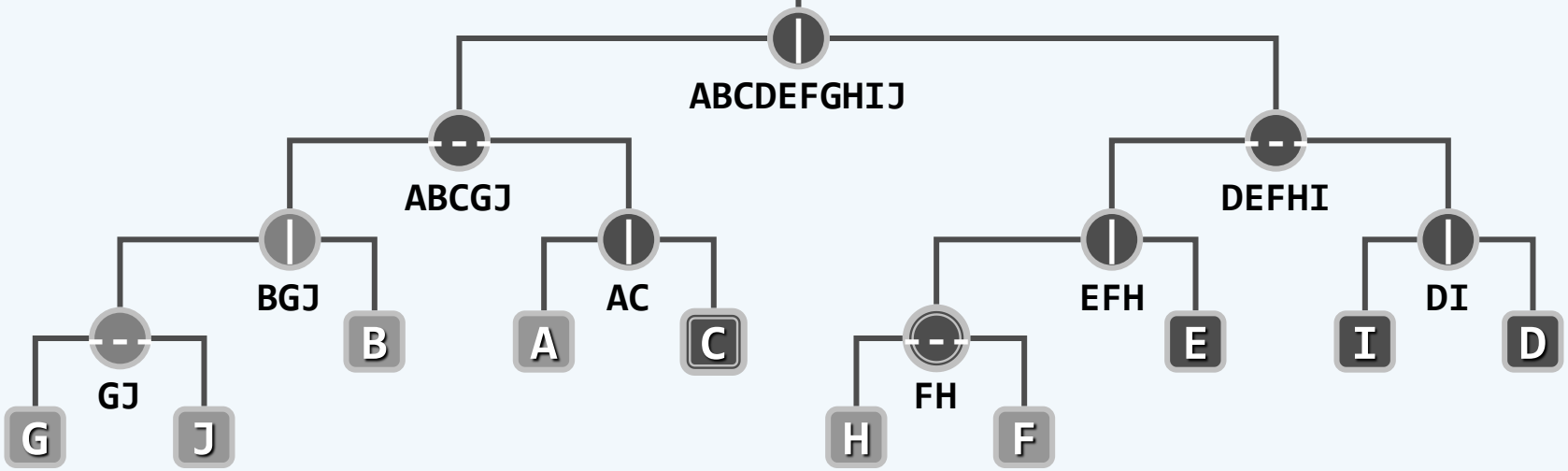





查找算法

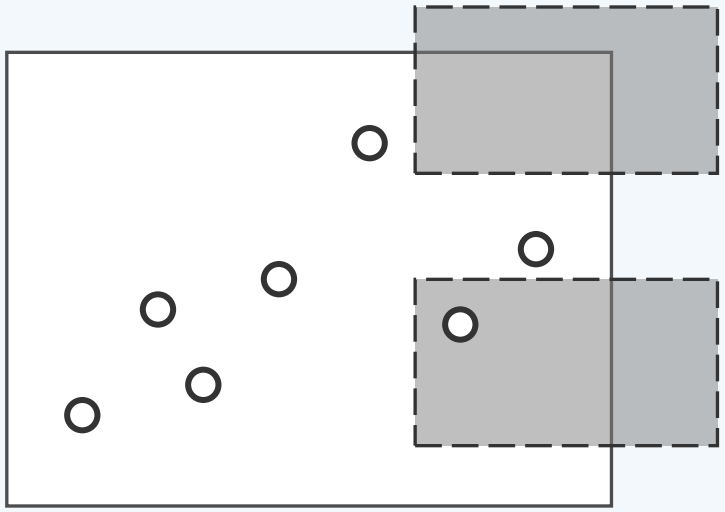
```
kdSearch( v, R ) {
   if ( isLeaf( v ) ) { if ( inside( v, R ) ) report( v ); return; }
   if ( region( v->lc ) ⊆ R ) reportSubtree( v->lc );
   else if ( intersect( region( v->lc ), R ) ) kdSearch( v->lc, R );
   if ( region( v->rc ) ⊆ R ) reportSubtree(v->rc);
   else if ( intersect( region( v->rc ), R ) ) kdSearch( v->rc, R );
}
```

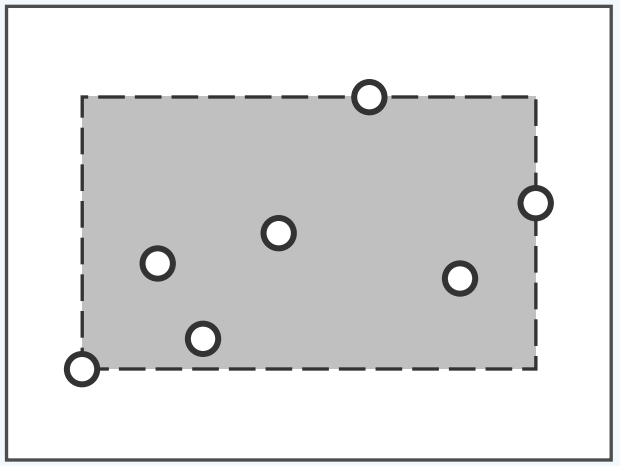
查找实例

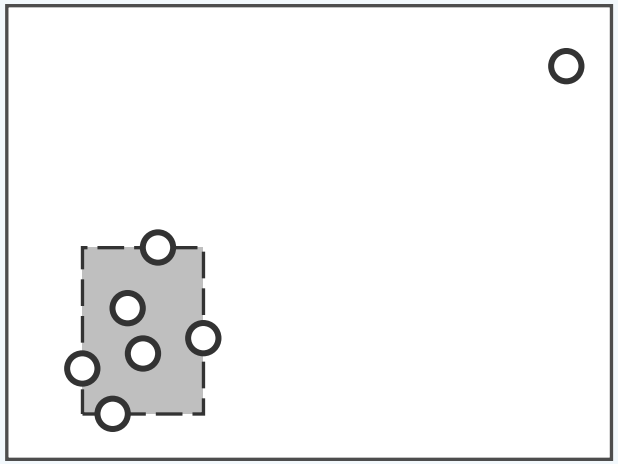

A
I
D
F
C
H
B
E
J
G

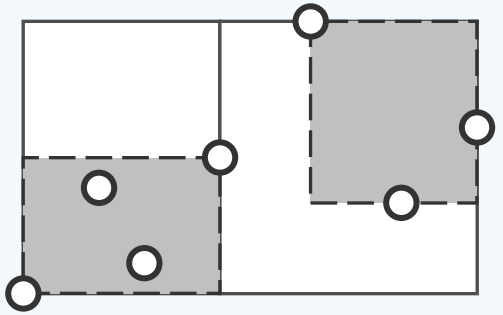

ABCDEFGHIJ
ABCGJ
DEFHI
BGJ
AC
EFH
DI
GJ
FH
G
J
B
A
C
H
F
E
I
D

包围盒









效率

- 空间　　$O(n)$

  预处理　　$O(n\log n)$

  查询　　$O(\sqrt{n} + r)$

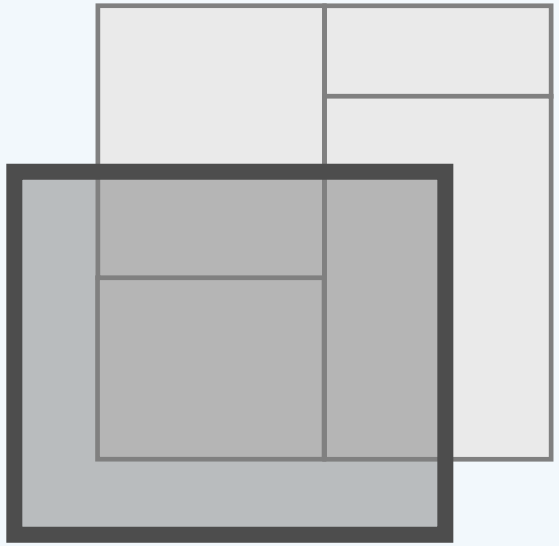
- 除去$O(r)$，其余查找时间决定于

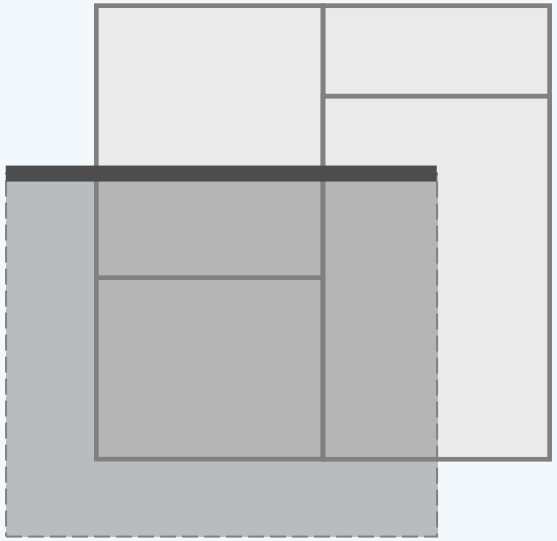
  递归发生的次数，或

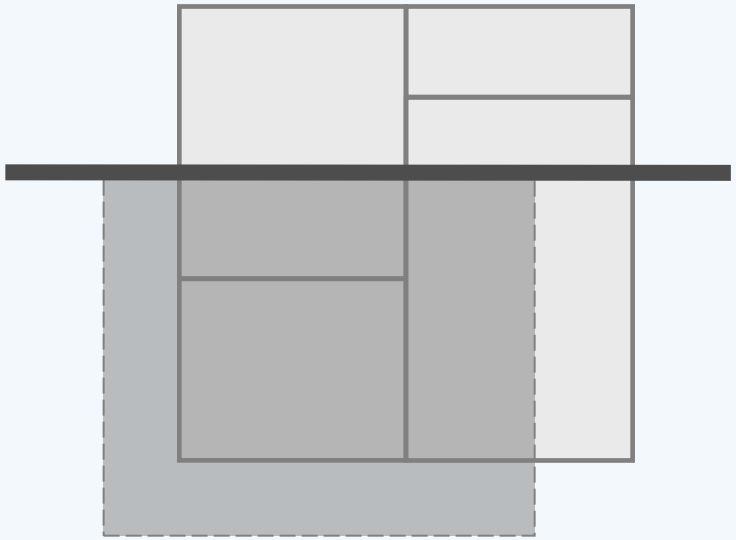
  待查找区域边界与子区域相交的数目$Q(n)$

- 递推关系：$Q(1) = O(1)$，$Q(n) = 2 + 2*Q(n/4)$

- 解得：$Q(n) = O(\sqrt{n})$







高维推广

- kd = k-dimensional
- 构造　深度 = 0 / 1 / ... / d-1时，沿第0 / 1 / ... / d-1维划分

  深度 = d / d+1 / ... / 2d-1时，沿第0 / 1 / ... / d-1维划分

  ...

  深度 = k时，沿第k % d维划分
- 空间：　$O(n)$
- 预处理：　$O(n\log n)$
- 查询：　$O(n^{1-1/d} + r)$

课后

- 在高维情况下，kd-树的效率如何评价？
- 如何消除（多点共垂直、水平线等）退化情况？
- 不借助median算法，如何构造2d-树？你所给算法的效率如何？
- 若只需计数，kd-树可在多少时间内给出解答？为此需对树做何调整？
- 适用于低（2~3）维数据的简化变种： quadtree, octree
- 针对三维以上的情况，更好的结构： multi-level search tree
  range tree
  interval tree
  priority search tree
  segment tree