

位置相关服务（LBS）： 基于R-树的查询

2017年

第五章 基于R-树的查询

1

范围查询

2

作业

3

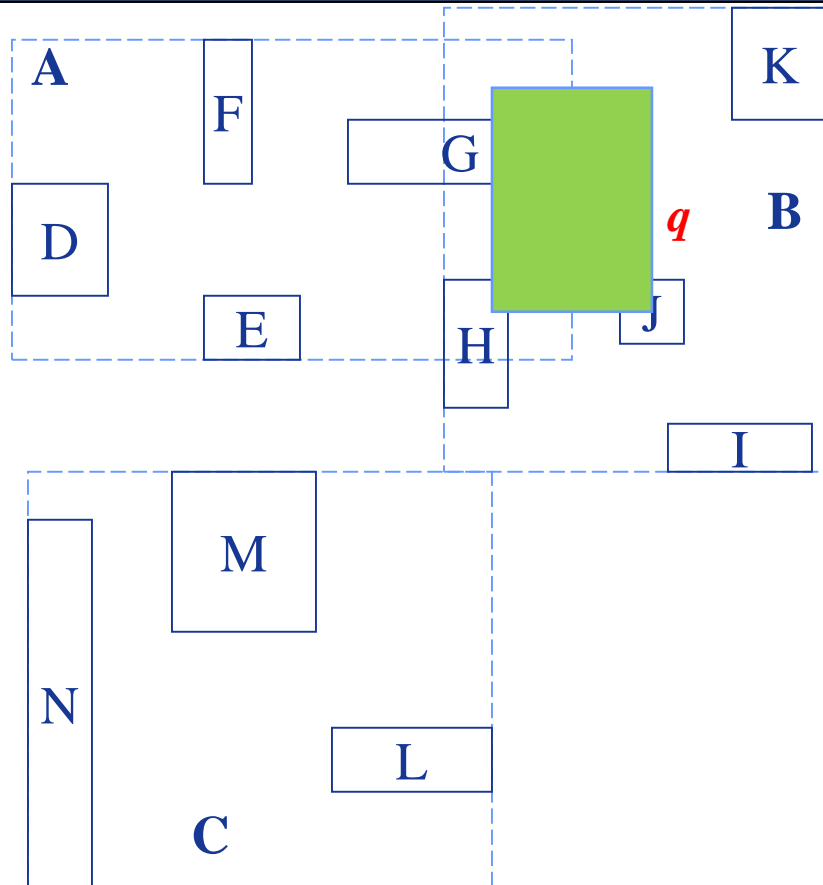
最近邻查询

4

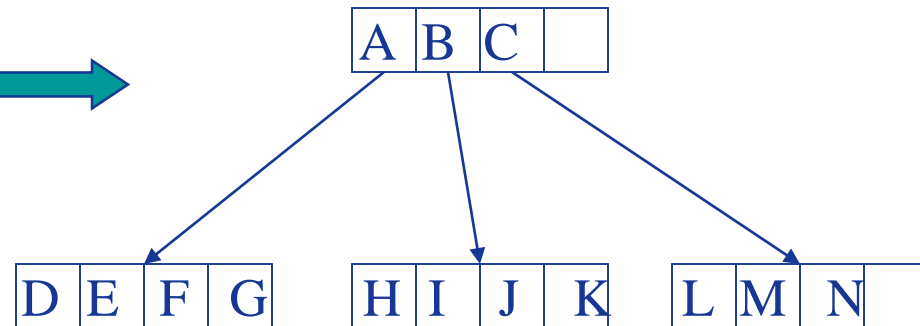
作业

基于R-tree的范围查询

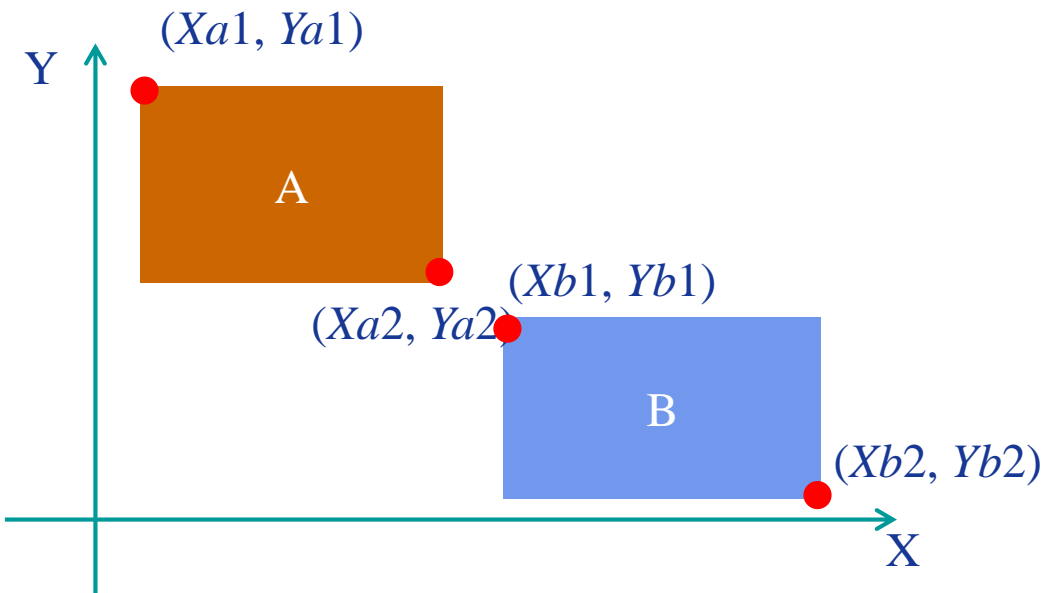
范围查询：对于给定的查询区域 q ，查询所有与 q 相交或包含于 q 中的数据。



从根节点开始，对所有内部节点，查找所有与 q 相交的项所指向的子节点；到达叶子节点后，输出所有与 q 相交的数据。



判断两个区域是否相交



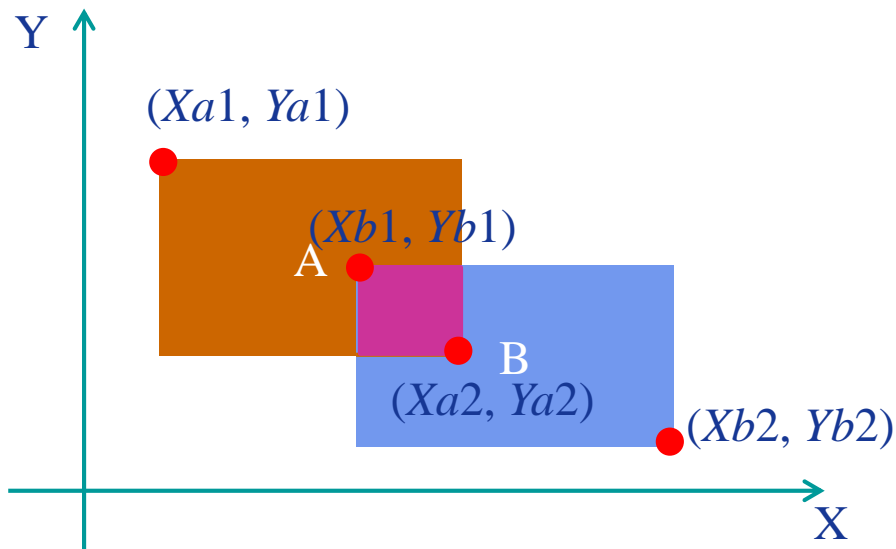
同时满足：

(1) X轴上：A、B范围的最大最小值 小于
A、B范围的最小最大值

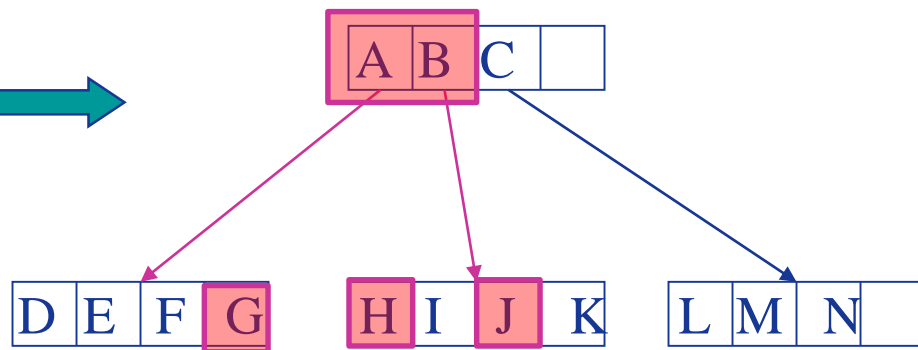
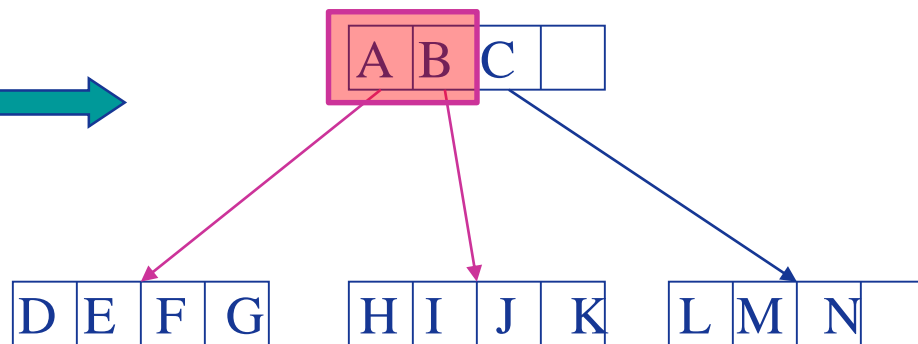
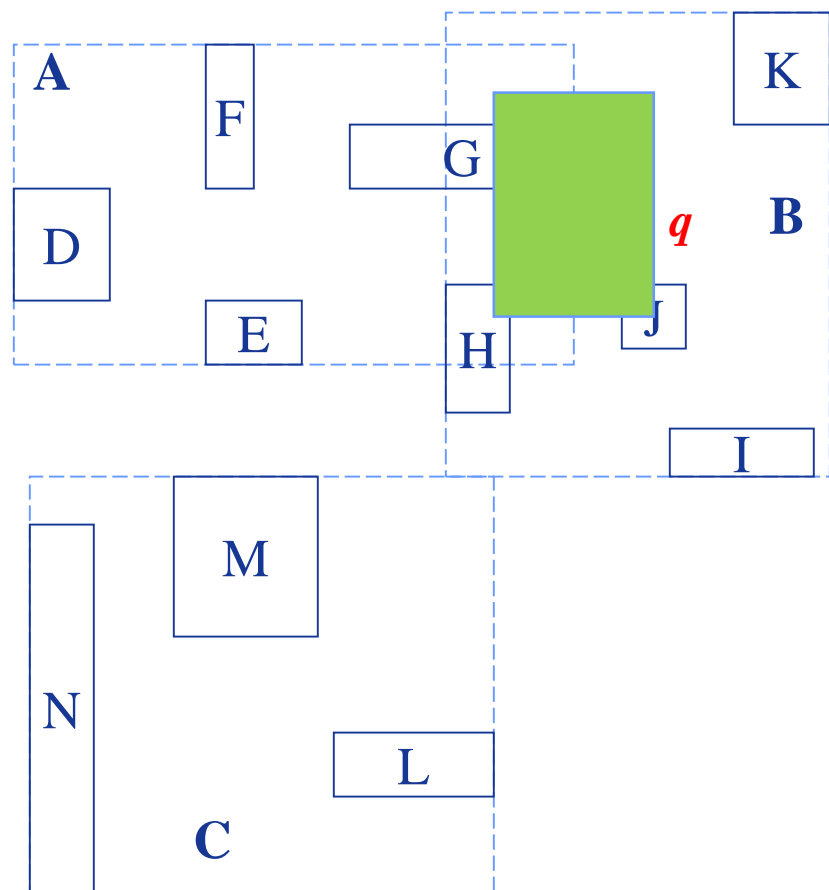
$$\text{即：} \text{Max}(Xa1, Xb1) < \text{Min}(Xa2, Xb2)$$

(2) Y轴上：A、B范围的最大最小值 小于
A、B范围的最小最大值

$$\text{即：} \text{Max}(Ya2, Yb2) < \text{Min}(Ya1, Yb1)$$



基于R-tree的范围查询



目 录

- 1 范围查询
- 2 作业
- 3 最近邻查询
- 4 作业

作业

❖ 实现基于R-树的范围查询

目 录

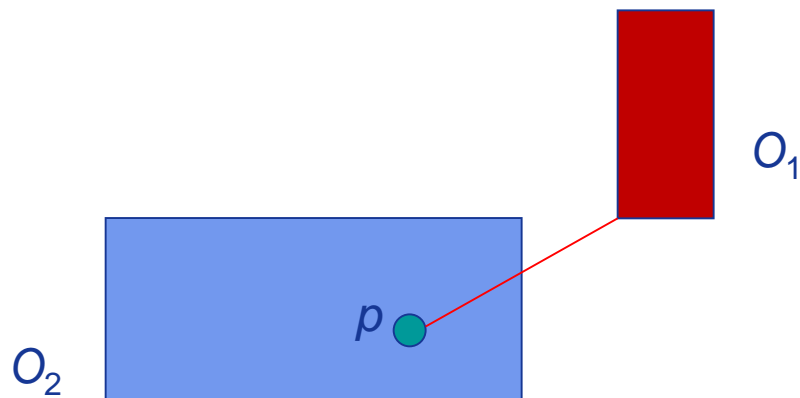
- 1 范围查询
- 2 作业
- 3 **最近邻查询**
- 4 作业

预备知识1:MinDist

P : 给定的查询点 O : R 树结点中的MBR

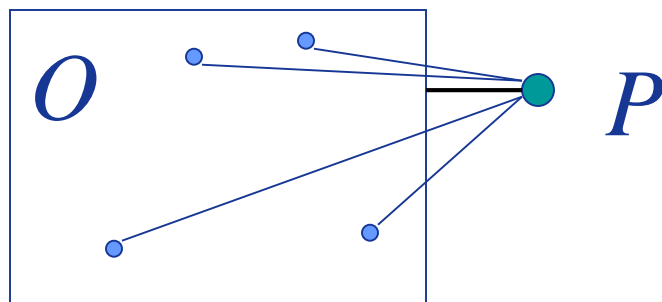
❖ 定义1：最小距离MinDist

- 如果点 P 在 O 内部，最小距离为0
- 如果点 P 在 O 外部，最小距离为 P 同 O 最近的一条边之间的距离



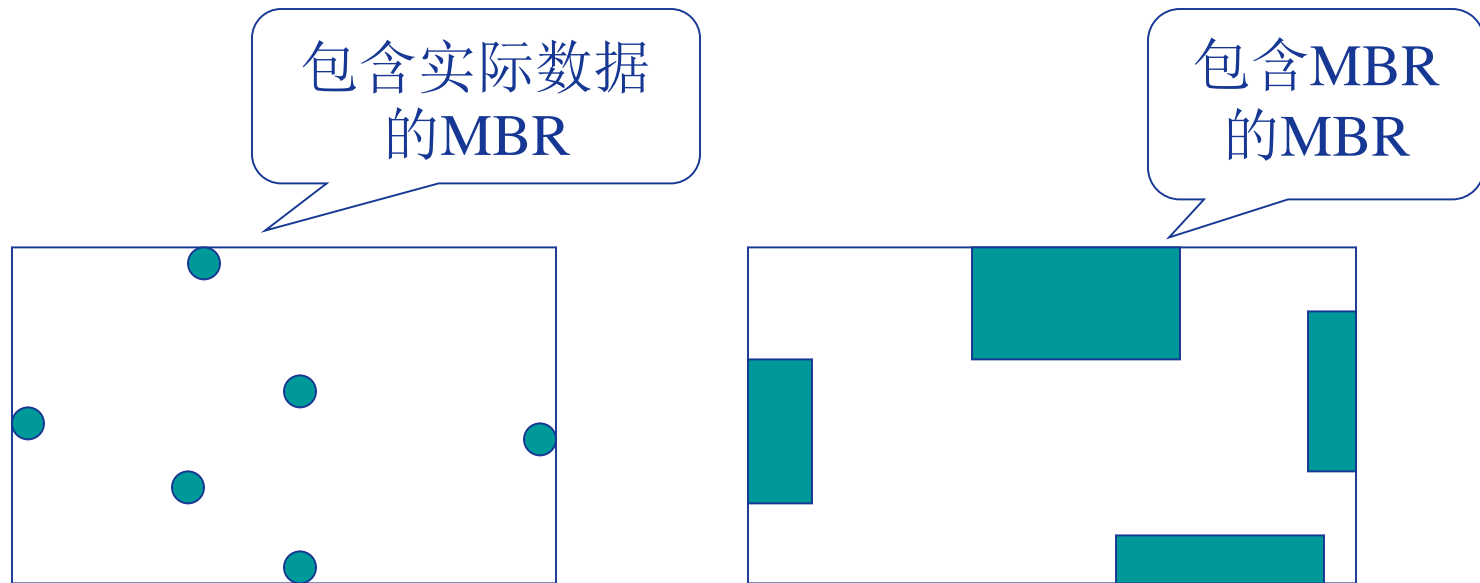
预备知识1:MinDist

- ❖ 推论1 : MinDist等于点 P 与 O 的边界上任意点之间的最小距离
- ❖ 定理1 : P 与 O 之间的MinDist小于或等于 P 同 O 中可能包含的所有数据之间的距离



预备知识2:MinMaxDist

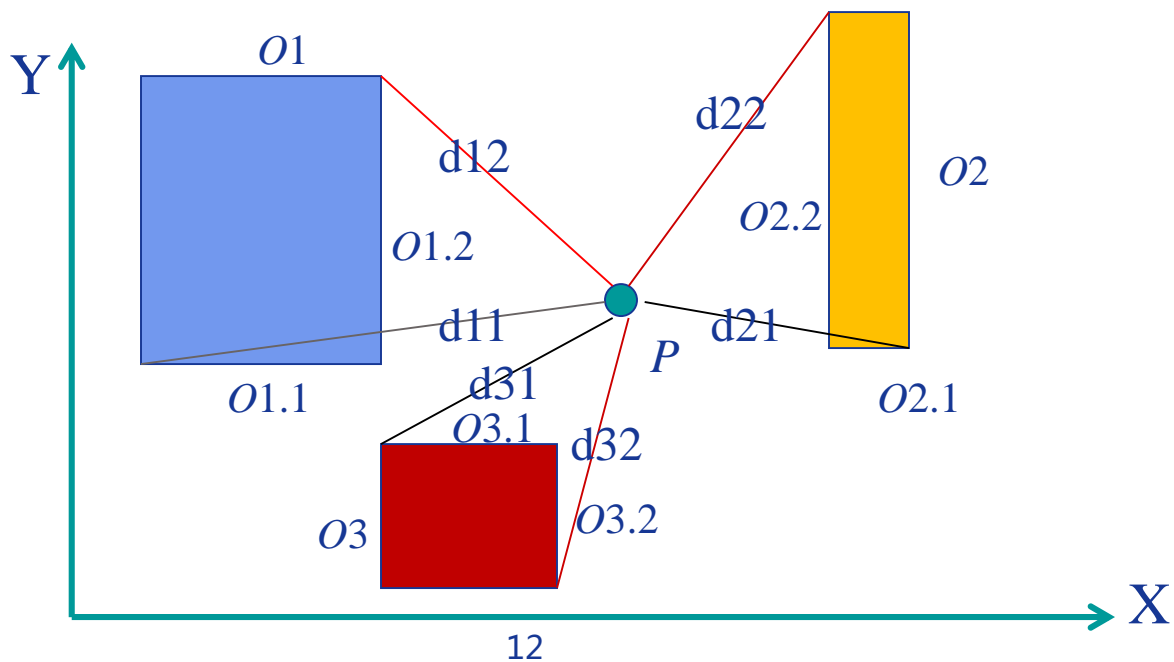
- ❖ 推论2：对于R树中的所有的MBR，它们的每一条边至少包含一个数据集中的数据。



预备知识2:MinMaxDist

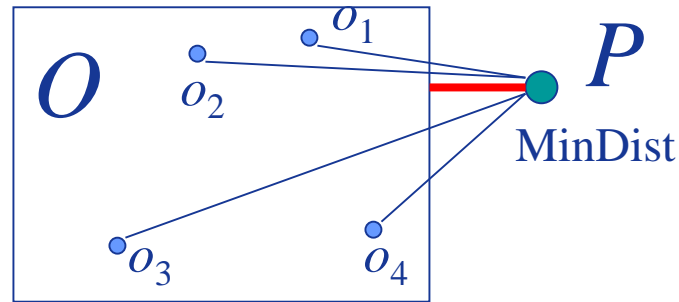
❖ 定义2：最小的最大可能距离MinMaxDist

- 对于每一维，求出该维中距离点 P 比较近的边上所有点中与 P 距离最远的那个点的距离
- 将所有维的距离中选择最小的那个距离定义为MinMaxDist

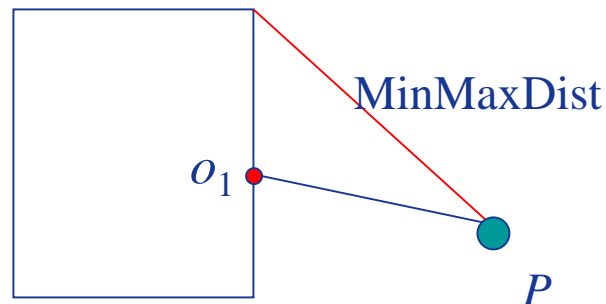


MinDist和MinMaxDist的意义

❖ O 中的所有数据与 P 之间的距离一定大于或等于MinDist

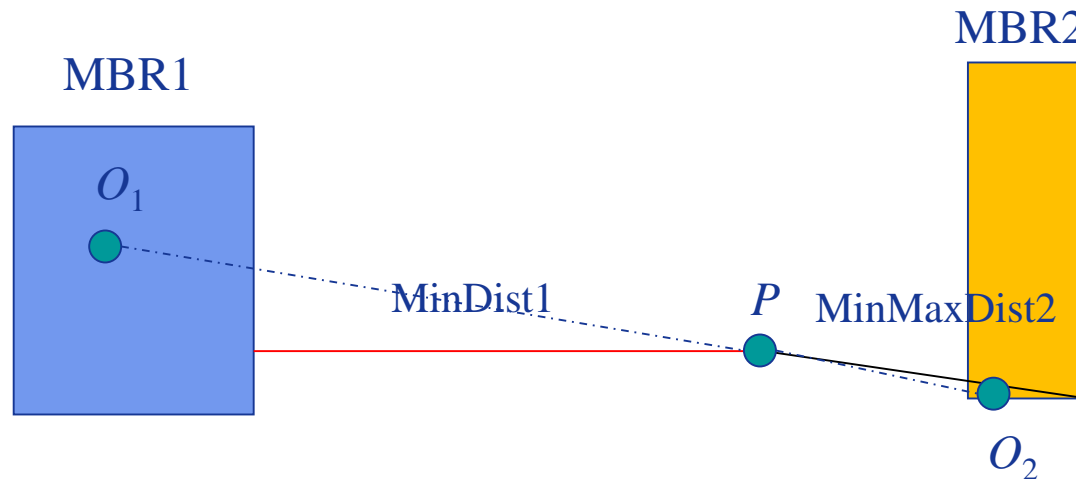


❖ O 中至少有一个数据与 P 之间的距离小于或等于MinMaxDist



剪枝策略1

- ❖ 若某一个MBR与 P 之间的MinDist 大于另外一个MBR与 P 之间的MinMaxDist,则可以放弃该MBR



证明:

(1) $\forall O_1 \in MBR_1, \text{dist}(P, O_1) > \text{MinDist1};$

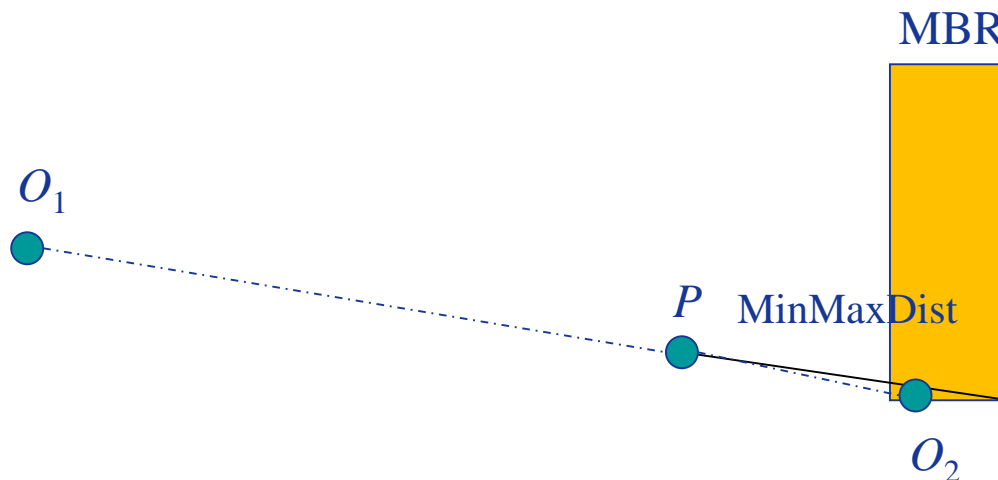
(2) $\exists O_2 \in MBR_2, \text{dist}(P, O_2) < \text{MinMaxDist2};$

那么, $\text{dist}(P, O_1) > \text{MinDist1} > \text{MinMaxDist2} > \text{dist}(P, O_2)$

所以, 放弃MBR1。

剪枝策略2

- ❖ 若某一个数据与 P 之间的距离大于一个MBR与 P 之间的 MinMaxDist , 则该数据可以放弃



证明:

(1) $\text{dist}(P, O_1) > \text{MinMaxDist}$;

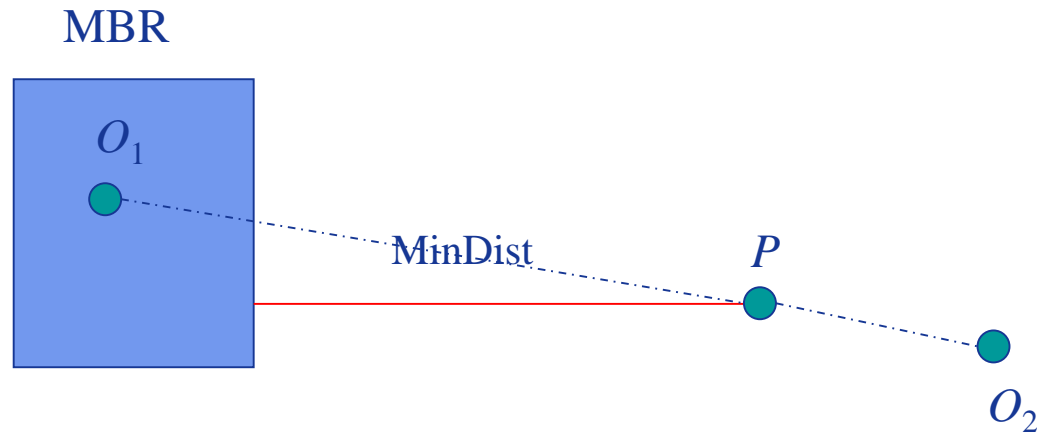
(2) $\exists O_2 \in \text{MBR}, \text{dist}(P, O_2) < \text{MinMaxDist}$;

那么, $\text{dist}(P, O_1) > \text{MinMaxDist} > \text{dist}(P, O_2)$

所以, 放弃 O_1 。

剪枝策略3

- ❖ 若某一个MBR与 P 之间的MinDist大于某一个数据与 P 之间的距离，则该MBR可以放弃



证明：

$$(1) \forall O_1 \in MBR_1, \text{dist}(P, O_1) > \text{MinDist};$$

$$(2) \text{dist}(P, O_2) < \text{MinDist};$$

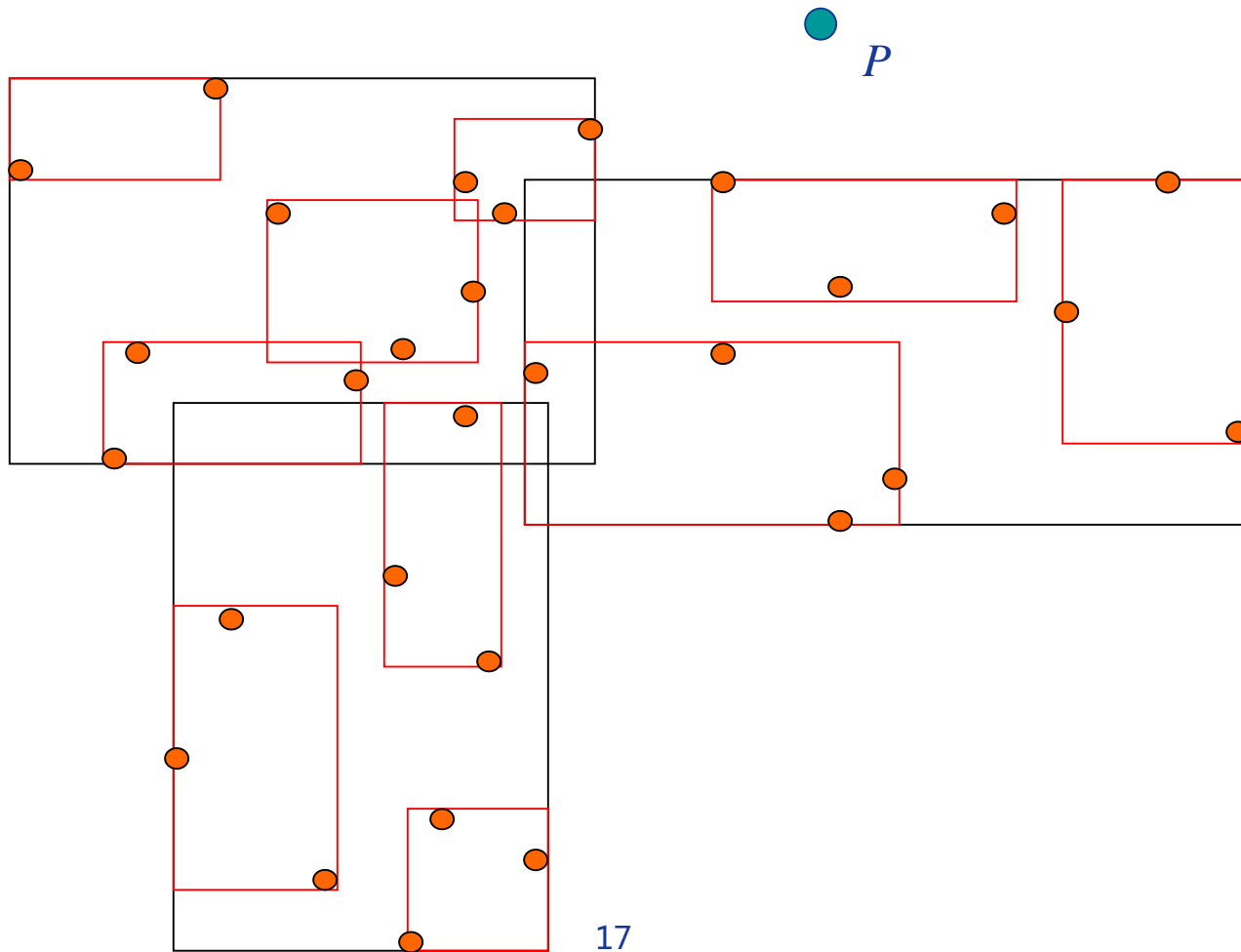
那么， $\text{dist}(P, O_1) > \text{MinDist} > \text{dist}(P, O_2)$

所以，放弃MBR。

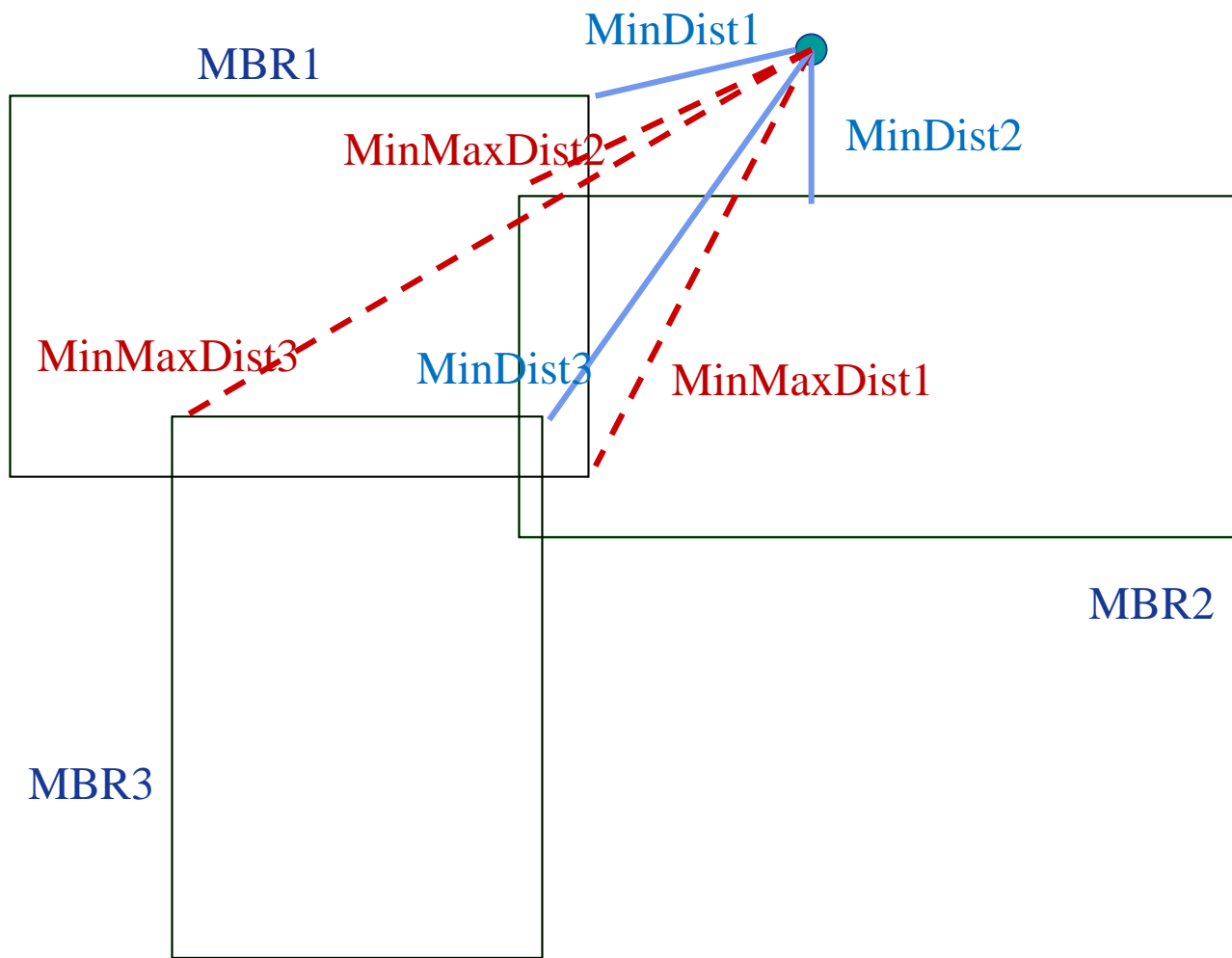
最近邻查询

基本思想：

从根节点开始深度优先遍历，采用MinDist和MinMaxDist进行剪枝。



最近邻查询



首先读入根结点

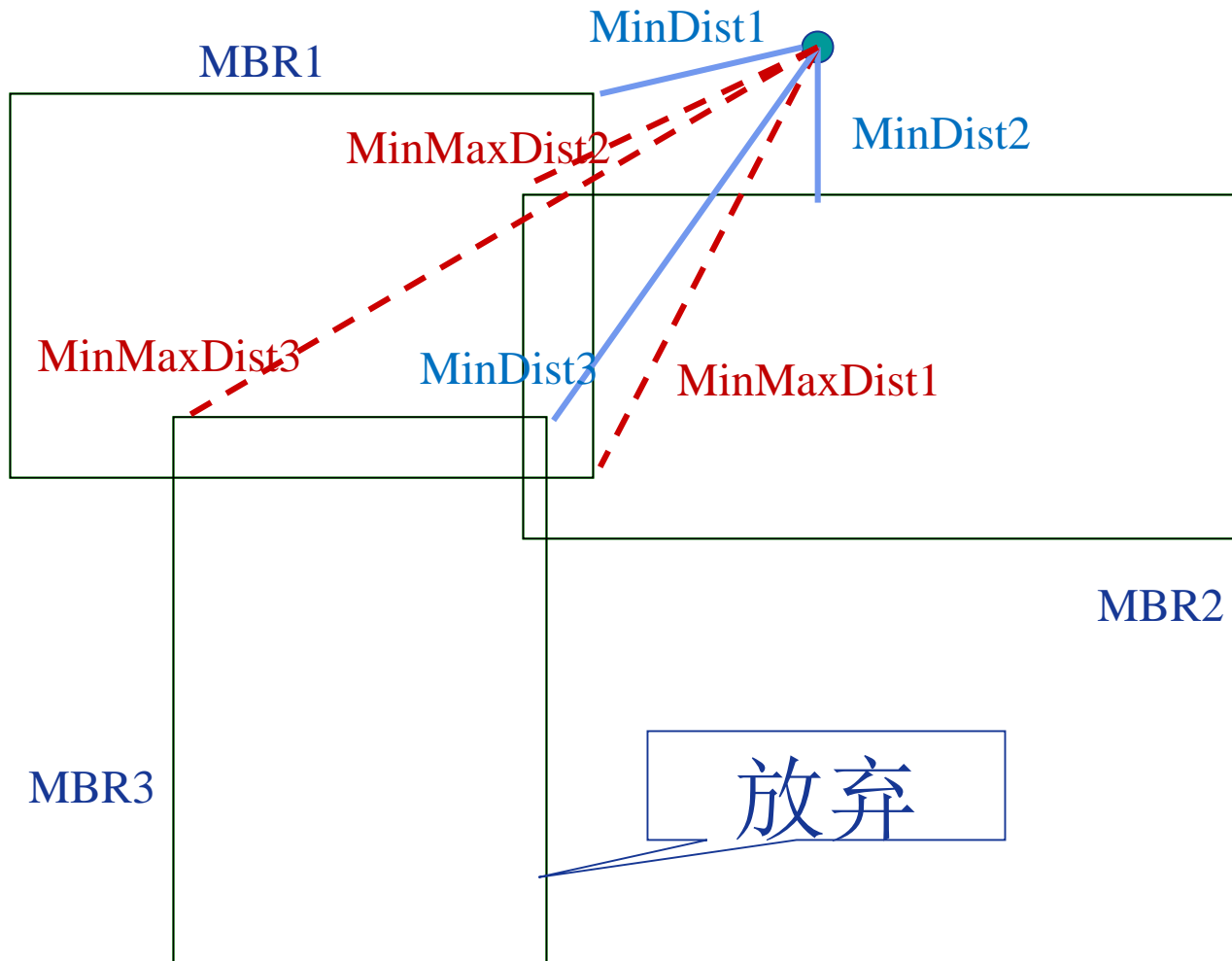
1.对三个MBR,按与 P 之间的MinDist排序。

2.按照刚才所讲的取舍原则
放弃不需要进行查找的MBR

3.对剩余的MBR,按照MinDist
的大小顺序对其对应的子结
点进行查询

4.该例子首先对右侧 (MBR2)
的那个结点进行查询

最近邻查询



因为:

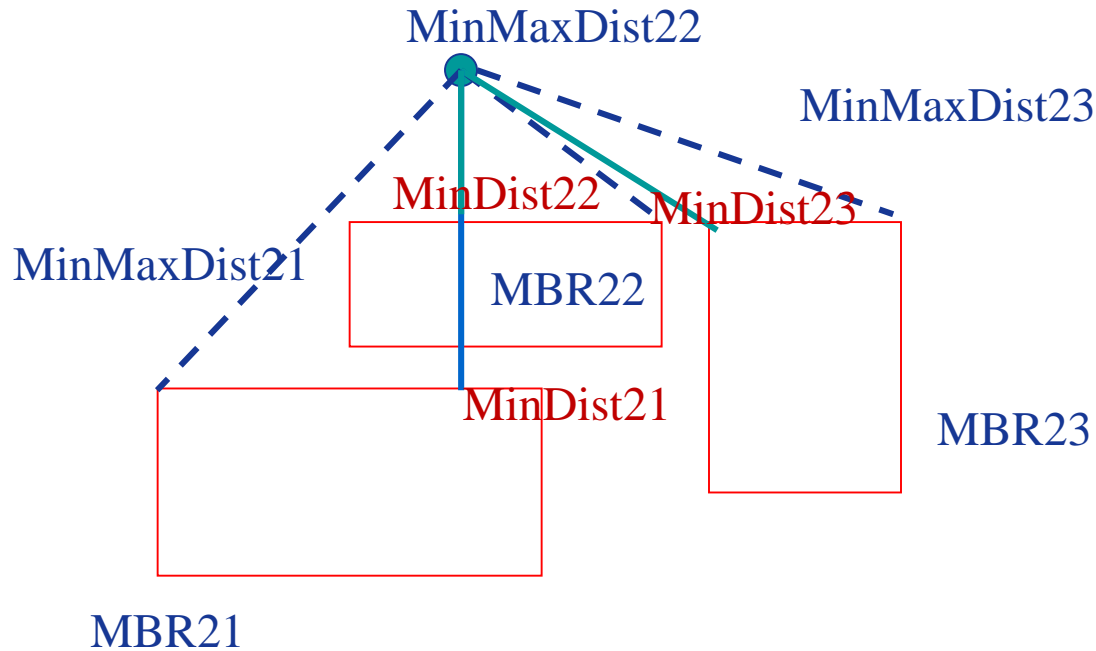
$\text{MinDist3} > \text{MinMaxDist2}$

所以:

依据剪枝策略1, 放弃

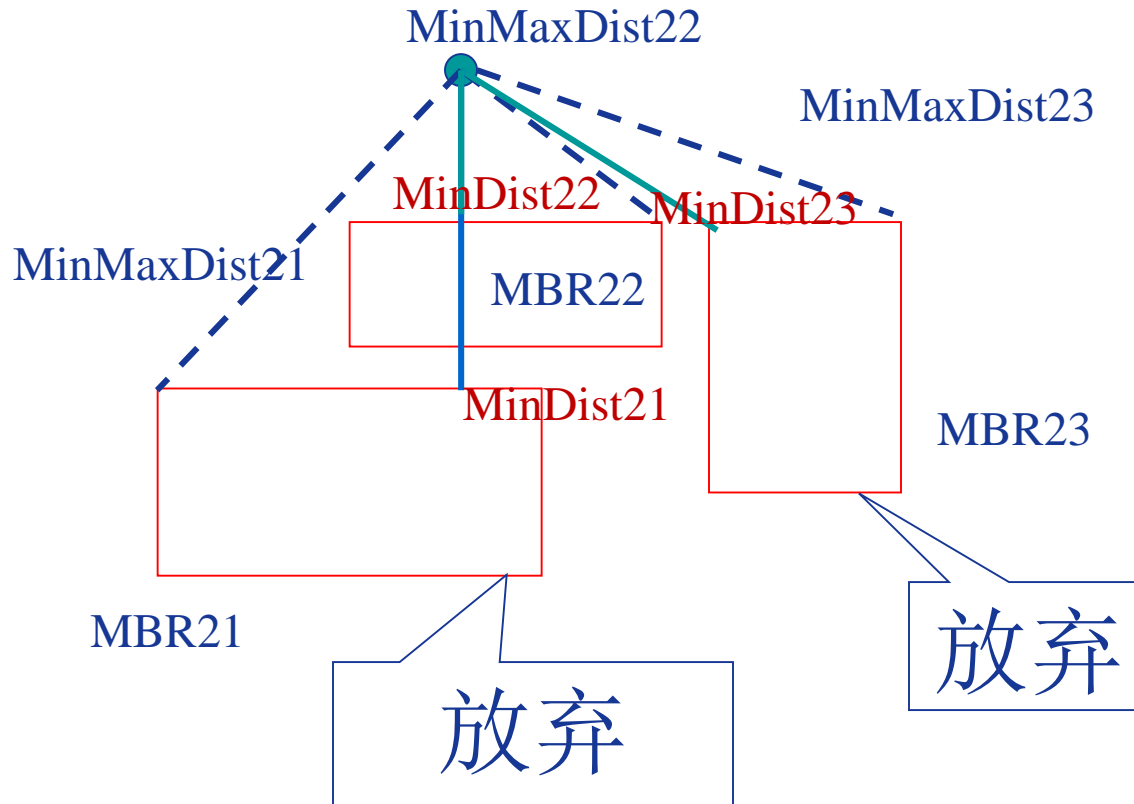
MBR3

最近邻查询



- 1.按刚才所述步骤对该结点中MBR进行取舍
- 2.对剩余的那个MBR对应的结点查询

最近邻查询



因为：

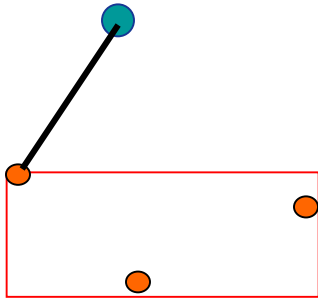
$MinDist21 > MinMaxDist22$

$MinDist23 > MinMaxDist22$

所以：

依据剪枝策略1，放弃MBR21
和MBR23

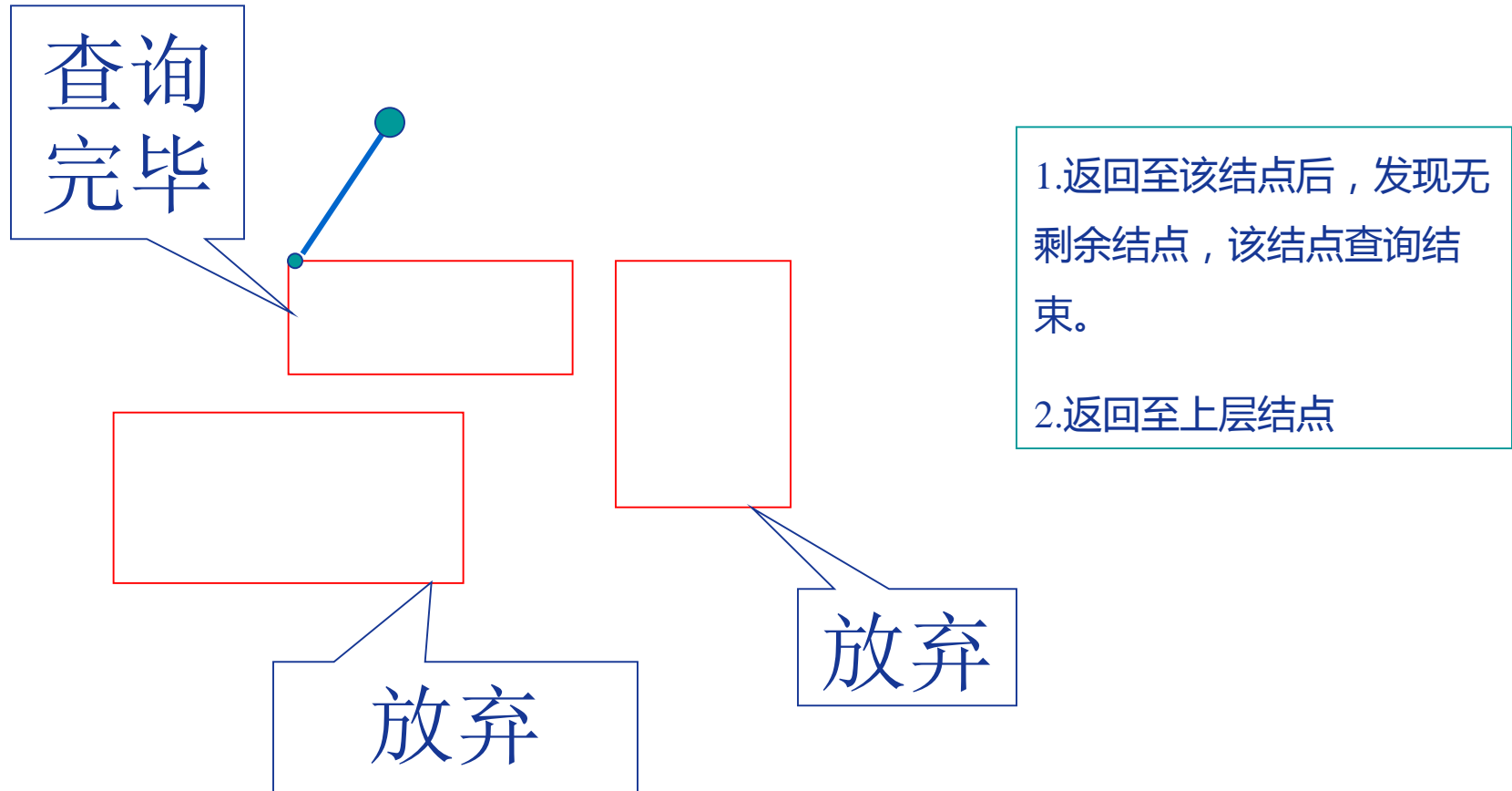
最近邻查询



1. 由于该结点是叶子结点，包含实际的数据，计算 P 同每个数据之间的距离，从中选取最小的那个距离，作为当前的最小距离，并将该数据设为当前的最近邻

2. 对该结点查询已经结束，返回上层结点（由于程序是递归调用）

最近邻查询



最近邻查询

放弃

1. 返回根结点。
2. 对剩余MBR继续调用取舍规则进行取舍，发现左上方的MBR应该放弃
3. 已无剩余MBR, 程序运行结束。

放弃

查询完毕

目 录

- 1 范围查询
- 2 作业
- 3 最近邻查询
- 4 作业

作业

❖ 实现基于R-树的最近邻查询

谢谢！

