Institut für Informationssysteme
Technische Universität Braunschweig

Technische Universität Braunschweig
Institut für Informationssysteme
http://www.ifis.cs.tu-bs.de

Karl Neumann, Sarah Tauscher

Spatial Databases and GIS

Solutions for Sheet 9

Exercise 1 (Spatial Join)

1. How do you have to modify the algorithm given below to join r-trees with different heights?
   If the r-trees have different heights, there will be "rectanglePairs" consisting of one leaf and one internal node. One possibility is to change FIND_INTERSECTING_PAIRS, so that it uses the leaf directly (and not its children) for the calculation. Another way is to write a new method (FIND_INTERSECTING_LEAVES) that compares a leaf to all children of a node and call this in an added "if" within the "foreach" loop (see below).

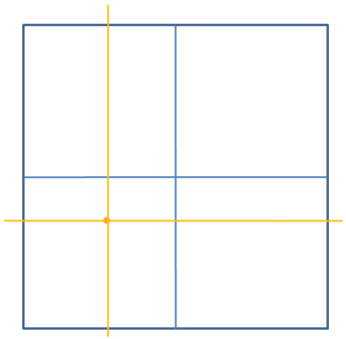
```
procedure FIND_INTERSECTING_LEAVES(leaf, node)
begin
 foreach child c of node do
  if l intersects c then
    if c is a leaf then
       p ← CREATE_PAIR(c, l)
       REPORT_INTERSECTIONS(p);
    else
     FIND_INTERSECTING_LEAVES(leaf, c)
    endif;
  endif;
 enddo;
end;
```

```
procedure INDEX_TRAVERSAL_SPATIAL_JOIN(rootA, rootB)
begin
  priorityQueue ← CREATE_PRIORITY_QUEUE();
  priorityQueue.ADD_PAIR(rootA, rootB);
  while NOT priorityQueue.EMPTY() do
    nodePair ← priorityQueue.POP();
    rectanglePairs ← FIND_INTERSECTING_PAIRS(nodePair);
    foreach p ∈ rectanglePairs do
      if p is a pair of leaves then
        REPORT_INTERSECTIONS(p);
      else if p consists of a leaf and a node then
       FIND_INTERSECTING_LEAVES(p.GET(0), p.GET(1))
       else
          priorityQueue.ADD_PAIR(p);
      endif; endif;
    enddo; enddo; end;
```

Technische Universität Braunschweig
Institut für Informationssysteme
http://www.ifis.cs.tu-bs.de

Karl Neumann, Sarah Tauscher

2. Develop a similar algorithm to join quadtrees.
   Basic idea: start with the root nodes, on the next level compare only the nodes being in quadrants that intersect. If the points are different, you'll always have to compare one quadrant with all 4 quadrants of the other tree (the quadrant which contains the point of the other tree), two with two quadrants of the other tree and one with only one quadrant, i.e. you have 9 (ne nw; ne ne; ne sw; ne se; nw nw; nw sw; se se; se sw; sw sw) comparisons instead of 16, but the runtime is still O(c$^{log\ n}$). Moreover, as data points are stored in internal nodes as well, you can't restrict the comparisons to nodes on the same level.



3. Can you think of a better strategy to join quadtrees?
   Search for the points of one quadtree in the other quadtree. Runtime O(n log n)