

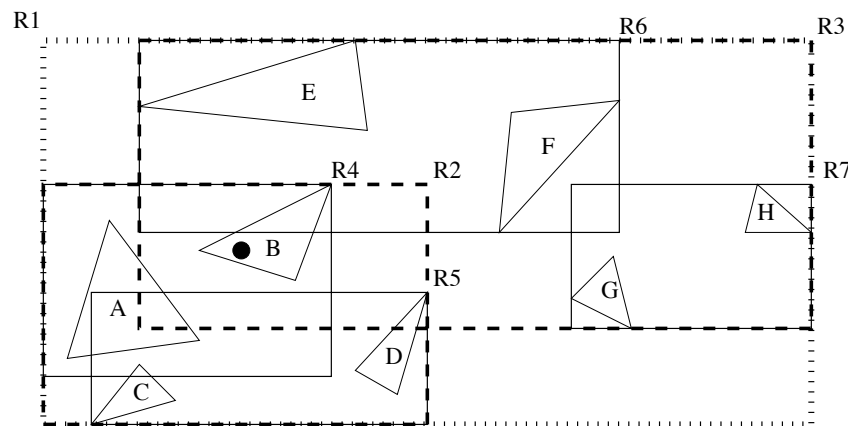
Database Tuning, ITU, Spring 2008

Rasmus Pagh

May 6, 2008

1 R-trees for triangles

In this problem we consider an R-tree that is used for storing non-overlapping triangles. We may draw a (binary) R-tree as follows:



Each rectangle corresponds to a node in the R-tree. In the drawing we have placed a name for each node at the upper right corner of its rectangle (except the root R1, whose name is at the top left corner). The leaf nodes containing the triangles are denoted A, B, C, D, E, F, G, H.

a) Consider a point query that asks for the triangle, if any, surrounding a point. In the example above, which nodes in the R-tree are visited when making the point query for the point shown as a black circle?

b) How many nodes in an R-tree with N leaves may need to be visited while performing a single point query? Give the worst example you can come up with, involving N triangles and 1 point. The example should be valid no matter how the rectangles of the R-tree are chosen.

In fact, there is a clever data structure based on persistent B-trees that gives an almost optimal solution to the above problem (a special case of “point location”). The drawback is that this data structure assumes the set of triangles to be static.