Uni Freiburg, Web Science Group
Prof. Peter Fischer
Systems Infrastructure for Data Science - Winter 2012/13

Exercise Sheet #3: Multi-Dimensional Indexing

Due/discussion: Nov 16, 2012

Exercise 3.1 : Key Concatenation

Instead of creating a multi-dimensional index, one idea is to use a conventional B$^+$-Tree where the key is the concatenation of the keys for each dimension. Suppose we create one such index on the following two attributes of a sales table: `month` and `salesperson` (in this order). Consider the following queries:

1. What amount has been sold by Al Bundy in September/2008?
2. How many soccer balls have been sold in July/2008?
3. How many soccer balls have been sold by Alice in all of our history?

Which of these queries may benefit from the index we have created? Explain your answer.

Exercise 3.2 : R-Tree, k-d Tree and Point Quad Tree

A. **R-Tree:** Consider an R-Tree with order d=2 where node capacity for such an R-Tree is 2d=4 and minimum occupancy is d=2. Assume that the database contains following rectangles:

Rectangle	Lower Corner (X,Y)	Upper Corner (X,Y)
1	(1,1)	(4,5)
2	(3,2)	(7,4)
3	(6,2)	(8,5)
4	(6,7)	(8,9)
5	(7,11)	(9,13)
6	(8,10)	(12,14)
7	(11,12)	(14,15)

The R-Tree for this database is shown in Figure 1:

Using the heuristics for insertion and node-splitting (*minimum necessary region enlargement* and *minimum totally covered area*) mentioned in the lecture, insert the following rectangles into the R-tree. Show the steps during insertion and draw the resulting R-tree.

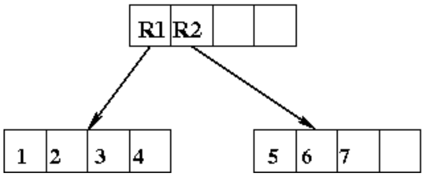

Figure 1: R-Tree with d=2

Rectangle	Lower Corner (X,Y)	Upper Corner (X,Y)
8	(6,3)	(9,6)
9	(4,10)	(6,13)

B. **k-d Tree:** Insert the following points in given order into an initially empty point k-d Tree:

(20,40), (10,50), (35,25), (30,20), (40,10), (25,15)

Show intermediate steps, the resulting k-d Tree and the two-dimensional partitioning figure.

C. **Point Quad Tree:** Repeat part B. for a point quad tree.