Group 9: Taylor Long & Agata Miszczyk

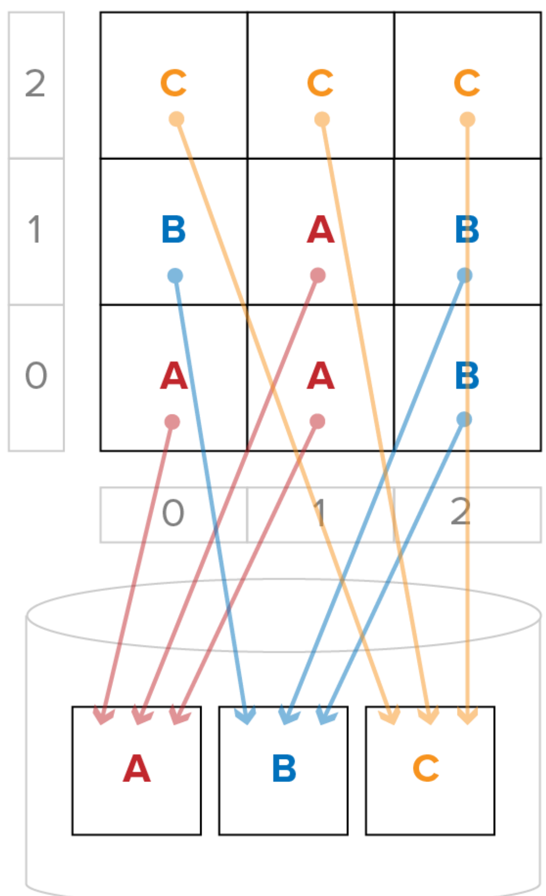
CSCI 5751: Homework 2

Our answers are italicized.

Chapter 4: Spatial Storage and Indexing

- **Q 4.5 :** Which of the following properties are true for R-trees, grid files, and B-trees with Z-order?
  (a) Balance (i.e., distance(root/directory, data-page pointer) is the same for all data-page points).
  (b) Fixed depth (i.e., distance(root/directory, data-page pointer) is a constant that is predefined).
  (c) Nonoverlapping: different nodes at a given level in an index are mutually disjoint.
  (d) 50 percent utilization per node: no node in an index is half empty.

- **Q 4.6 :** Draw a final grid file structure generated by inserting the following sequence of points. Assume each disk page can hold at most three records.
  (0,0), (1,0), (1,1), (0,1), (2,0), (2,1), (2,2), (1,2), (0,2)
  Clearly show the dimension vector, directory grids, and data page.



- **Q 4.16 :** Review the R-tree in Figure 4.16. Which nodes are accessed for
  (a) a point query for a point inside rectangle 4.
  R
  x
  c
  (b) a range query for minimal orthogonal rectangle including rectangle 7 and 10.
  R,
  x, y
  b, c, d , e, f

Chapter 5: Query Optimization

- **Q 5.1 :** For the given data space consisting of nine octahedrons (see Figure on Page 147), calculate the cost of processing a range query:
  (a) Directly

  $$Pi(Q) = \sum_{i=4}^{9} 8n$$

  = 48

  Since we are assuming the cost of processing a polygon is represented by the number of edges (8), our equation is taking the 6 octahedrons that intersect or are adjacent to A and muliplying them by 8.

  (b) Using the filter-refine strategy

  $$Pi(Q) = \sum_{i=4}^{9} 4n$$

  = 24

  When we assume that the filter step is drawing MBRs around each octahedron, the number of sides is 4. This cuts the processing cost in half.

  Assume there is no index on the space and the cost of processing each polygon is equal to the number of edges. For the filter step, assume each polygon is represented by its MBR.

- **Q 5.4 :** Consider the following strategies for spatial-join tree matching, nested loop with index, choose applicable strategies for a spatial join under the following scenarios:

(a) Overlap join: neither data-set has available R-tree indices

- *Nested Loop - cost can be prohibitive*

(b) Overlap join: one data-set has available R-tree indices

- *Nested Loop with Index - do no need to scan inner relation completely at each iteration*

(c) Overlap join: both data-sets have available R-tree indices

- *Tree Matching*