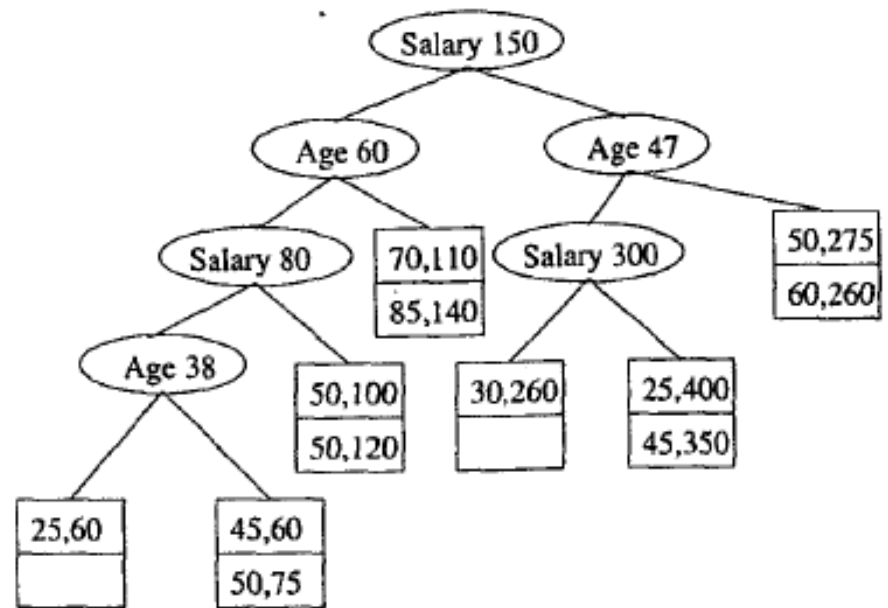


Trees for multidimensional data

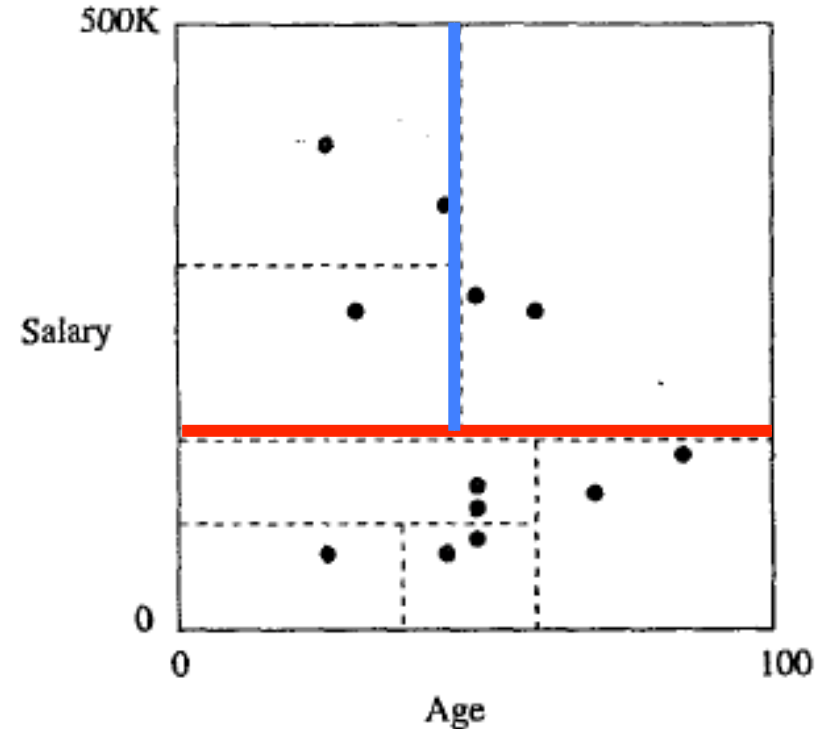
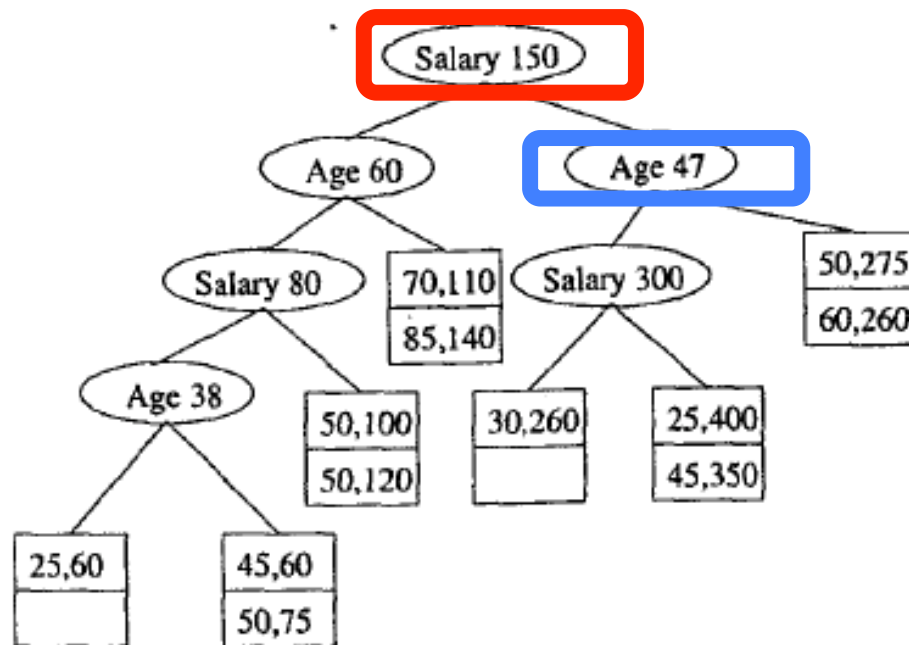
- There are dozens of such structures
- **kd-Trees**
 - k-dimensional search tree
 - Intuition: a binary search tree to multidimensional data where each dimension has values interleaved along paths from root to leaves
- **kd-Tree node**
 - have an attribute **a** with an associated value **V**
 - node splits data points into those with a-value $\leq V$ and a-value $> V$
 - attributes **a** at different levels of the tree **are different!**

kd-Tree example

- Classical kd-Trees have values within interior nodes...
- ... but our database-adapted kd-Tree restricts values to leaf nodes...
- ... that is, each interior node is in memory & leaf nodes are disk blocks
- Example:
 - First level: salary
 - Second level: age
 - Other levels: continue to alternate
- In effect we are partitioning the data



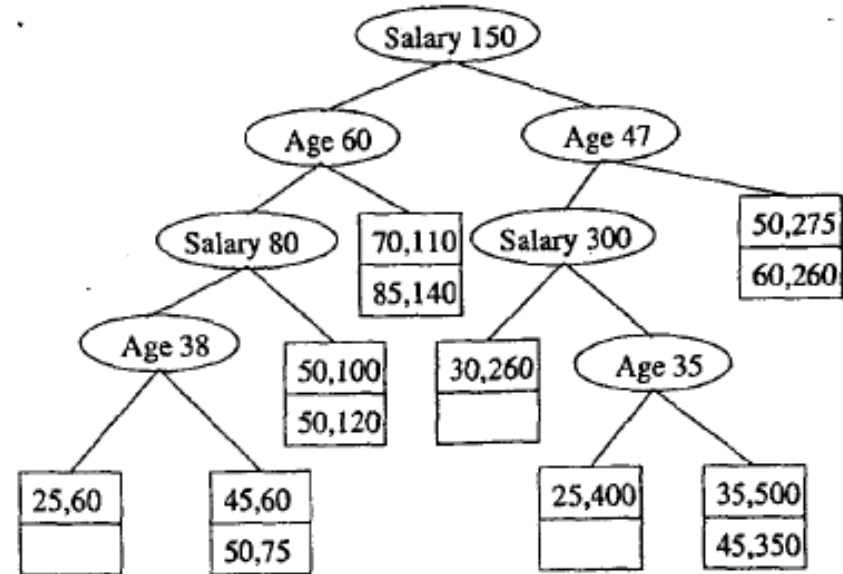
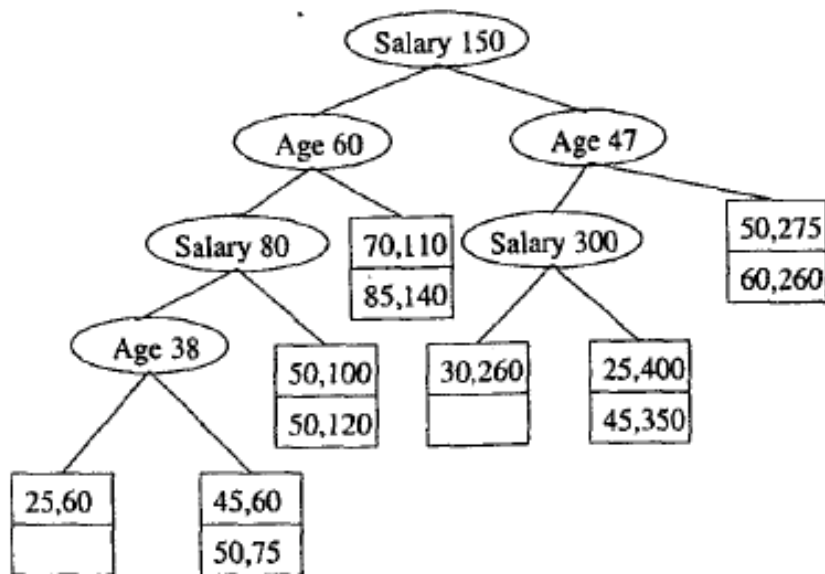
kd-Tree corresponding partitioning



kd-Tree operations

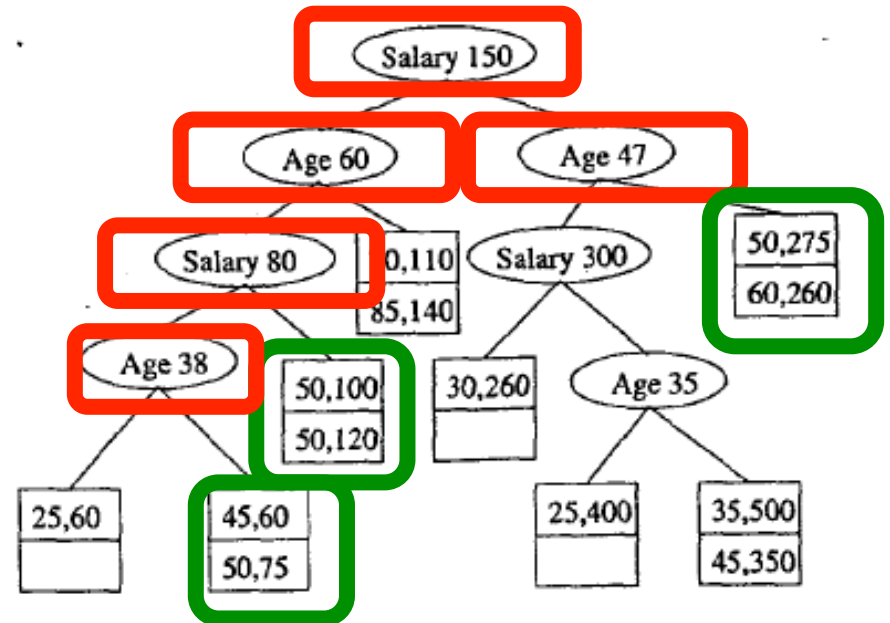
- Lookup
 - Just like a binary search tree, albeit with lookup based on the attribute at each relevant level
- Insertion
 - Start with lookup
 - If we reach a block with room, the insert into spare slot
 - Otherwise split block in two, dividing according to appropriate attribute for new level (and creating appropriate interior node)
- Example: insert data point for 35-year-old gamer with salary of \$500K.

Insert: 35-year-old gamer; \$500K



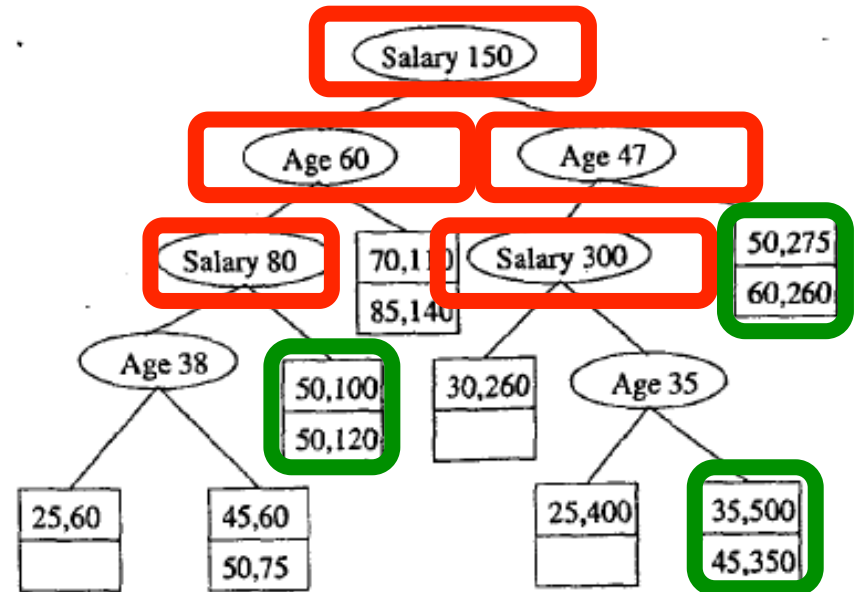
kd-Tree operations

- Partial-Match queries
 - Recall this query corresponds to having values for some attributes but not for others
- Example: Return all points with age = 50.
 - Sometimes we must look at both children of an interior node...
 - ... while at other times we need only look at left or right subtree.



kd-Tree operations

- Range Queries
 - Range might clearly indicate one child node...
 - ... but may just as easily straddle a node.
- Example:
 - Ages: 35 to 55
 - Salaries: \$100K to \$200K



Colophon

- Some diagrams taken from first edition of "Database Systems: The Complete Book"
- A bit of the B+ tree algorithm material based on <http://dblab.cs.toronto.edu/courses/443/2013/05.btree-index.html>
- Additional concepts on multidimensional indexes: "Multidimensional Access Methods", Volker Graede & Oliver Günther (ACM Computing Surveys, 30(2), pp. 170-231)