EC504 Project Proposal

Add names of and detail of project plan as far as you know

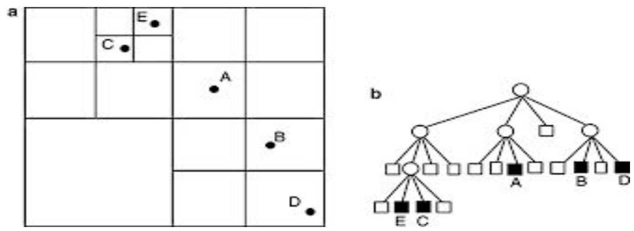
Boston University School/college name here

Proposed Data Structures and Implementations

- Spatial Data Structures
  - R trees
  - KD trees
  - Octrees
  - Quadtrees
- Linear Data Structures
  - Fusion Tree
  - Van Emde Boas Tree

Quadtrees

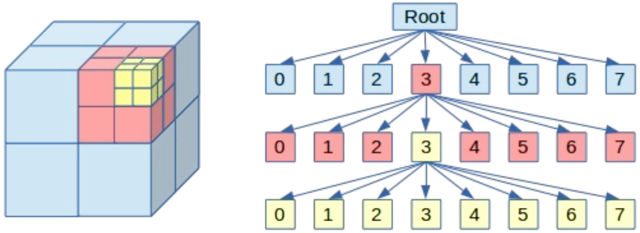
- Tree data structure with each node containing exactly 4 child
- Used to represent data in 2D
- Divides regions into quadrants
- Proposed implementation: Node structure for a point object structure, linear object structures
- Proposed implementations: space decomposition, Insertion, deletion, key search



Source: Vassilakopoulos M., Tzouramanis T. (2018) Quadtrees (and Family). In: Liu L., Özsu M.T. (eds) Encyclopedia of Database Systems. Springer, New York, NY

Octrees

- Tree data structure with each node containing exactly 8 child
- Used to represent 3D space
- Divides 3D space (cubes) in octants
- Used for space partitioning , color quantification, finite element analysis, 3D mapping state estimation
- Proposed implementation: spatial partitioning, neighbour key search, insertion, deletion



Source : https://geidav.wordpress.com/2014/07/18/advanced-octrees-1-preliminaries-insertion-strategies-and-max-tree-depth/

BOSTON UNIVERSITY

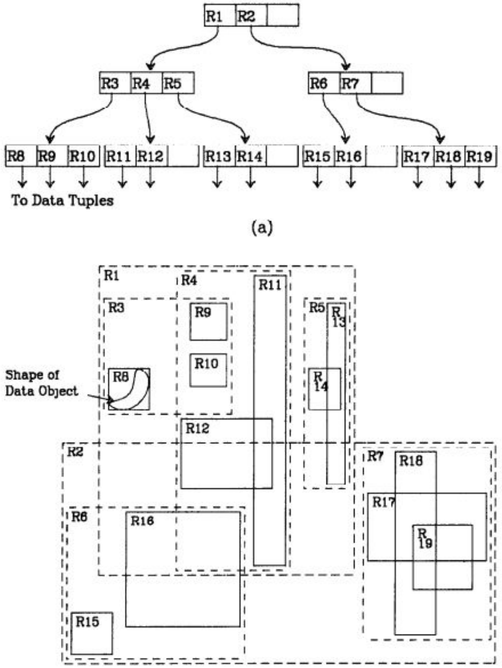
R-Trees

- Multiple entries per node (similar to B-Tree).
- Leaf nodes contain information related to spatial objects: *(I, tuple-identifier)*
  - $I = (I_0, \ldots, I_d)$ where I is length in dimension i
  - tuple-identifier = $\boldsymbol{x} = [x_1, \ldots, x_d]$, or a spatial coordinate.
  - Think of a "bounding box."
- Internal nodes: (I, child-pointers).
- Fallback Goal: find node (exact match), insert node, delete node, update node
- Ideal Goal: NN Query

k-d Trees

- Multiple entries per node
- Each node contains:
  - Keys: $\boldsymbol{x} = [x_1, \ldots, x_d]$, or a spatial coordinate
  - DISC(node) = index to key that separates low subtree (left) from high subtree (right) by comparing corresponding "coordinate" in children.
  - Pointers to low and high subtrees.
- Nodes of equal depth have the same DISC().
- Used for space partitioning , color quantification, finite element analysis, 3D mapping state estimation
- Fallback Goal: find node (exact match), insert node, delete node, update node
- Ideal Goal: NN Query

R-Trees



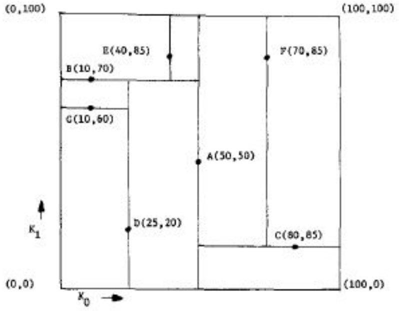
(b)
Figure 3 1

Guttman, A. (1984). "R-Trees: A Dynamic Index Structure for Spatial Searching" (PDF). *Proceedings of the 1984 ACM SIGMOD international conference on Management of data – SIGMOD '84*. p. 47. doi:10.1145/602259.602266. ISBN 978-0897911283.

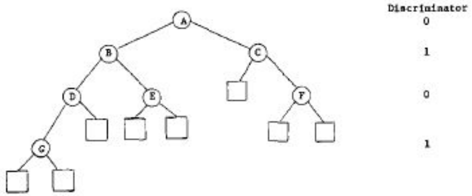
Boston University School/college name here

k-d Trees

Fig. 1. Records in 2-space stored as nodes in a 2-d tree.
Records in 2-space stored as nodes in a 2-d tree (boxes represent range of subtree):



Planar graph representation of the same 2-d tree (*LOSON*'s are expressed by left branches, *HISON*'s by right branches, and null sons by boxes):

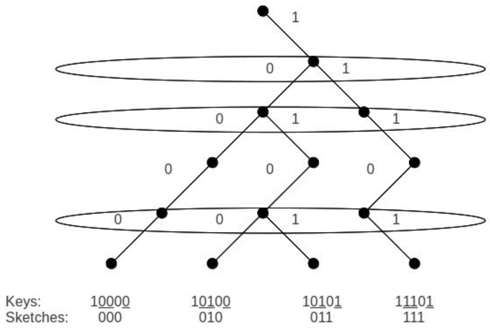


Bentley, J. L. (1975). "Multidimensional binary search trees used for associative searching". *Communications of the ACM*. **18** (9): 509–517. doi:10.1145/361002.361007

BOSTON UNIVERSITY

Fusion Trees

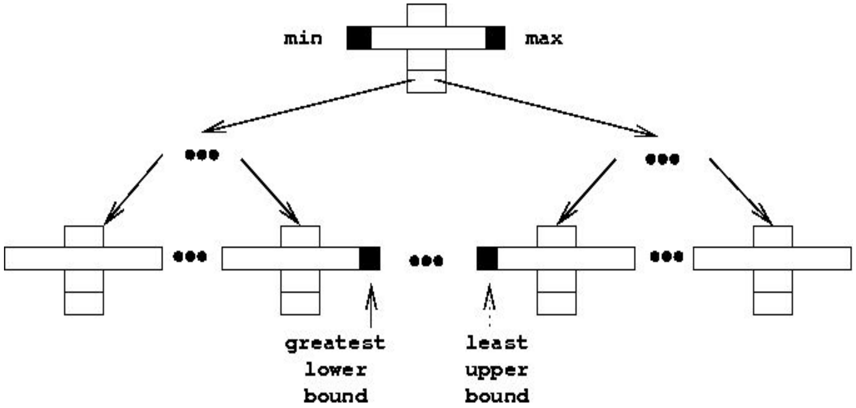
- a **fusion tree** is a type of tree data structure that implements an associative array on *w*-bit integers
- Using a technique called sketching and desketching, it solved finding the predecessor and successor problem in O(1) time with kO(1) preprocessing
- An application of fusion trees to hash tables



Boston University School/college name here

Van Emde Boas Tree

- Similar to fusion tree, a van emde boas tree also implements an associative array on a w-bit integer
- It performs all operations in $O(\log m)$ time, or equivalently in $O(\log \log M)$ time, where $M = 2^m$ is the maximum number of elements that can be stored in the tree
- The root node has all the elements of the universe with min max value and every node below it has root of its parent's elements.



Implementation Details

- Language of implementation: C++
- Tests and analysis using synthetic datasets: large and small sized input as well as “edge cases” that are unique to each data structures
- Analysis will consist of amortized opcount to look at average cost for the *basic* functions for each data structure: find node, insert node, delete node, update node
- Command line interface for user-input queries.