Comparative Analysis of Data Structures

Andres Chavez, John Neal

BOSTON UNIVERSITY

kd-Trees Implementation

- Each node contains:
  - Coordinates
  - Pointer to right subtree
  - Pointer to left subtree
  - discriminator (index of coordinate that splits right and left subtrees)
- kd_tree class contains:
  - pointer to root
  - d (dimension of coordinates in tree)
  - Methods:
    - void insert_node(int* coordinates)
    - void delete_node(int* coordinates)
    - Node* find_node(int* coordinates)

BU

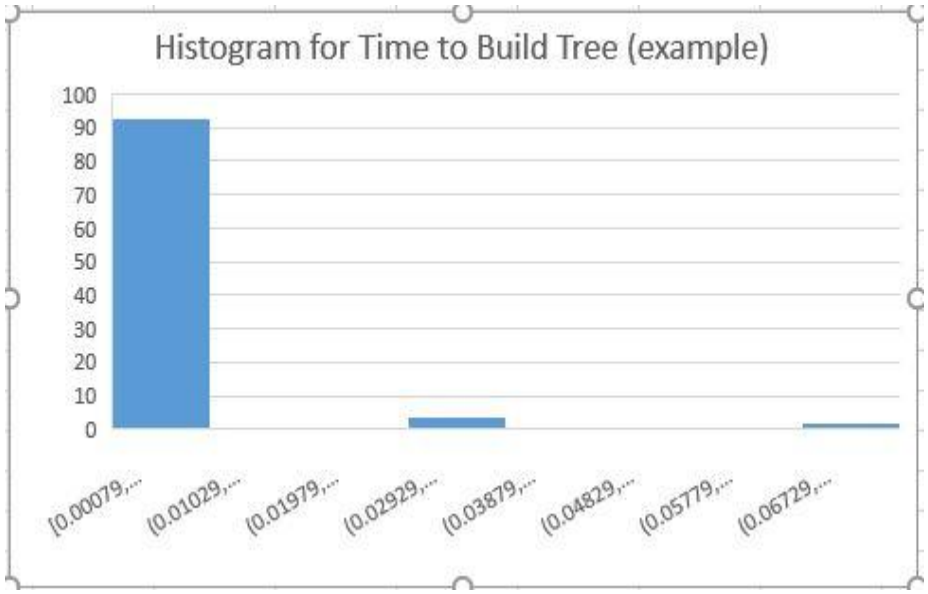
Random Datasets

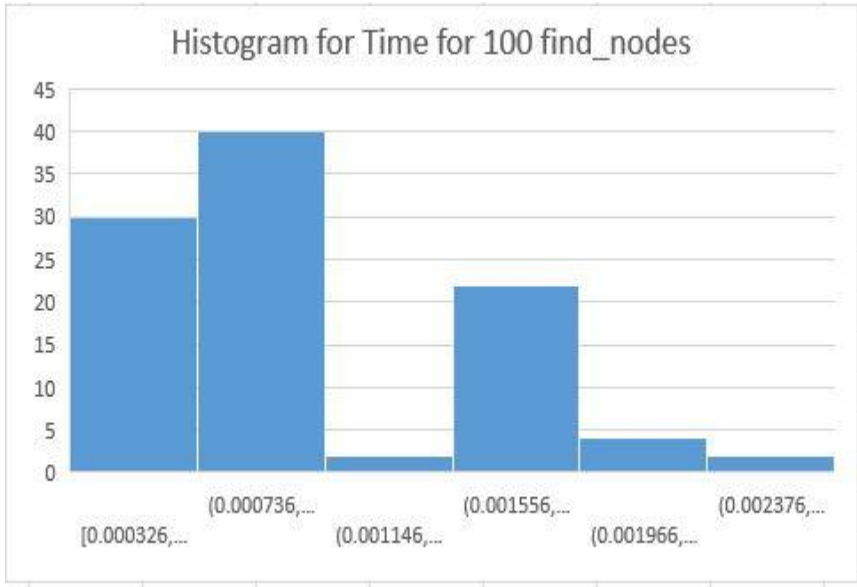
- Random Datasets
  - 100 cases for n = 100, 1000 and 10 cases for n = 100K, 1M
  - sampled from uniform[0, n*10]
  - Goal: Generate histogram of run times for find, insert, delete and compute average runtime for each n. (Varying data)
- Random Dataset Permuted
  - 100 cases for n = 100, 1000 and 10 cases for n = 100K, 1M
  - Permutations of 0th random dataset for each n-case
  - Goal: Generate histogram of run times for find, insert, delete and compute average runtime for each n. (Varying shapes)

Results

- find_node, insert_node, delete_node work for Handwritten_Test_Cases
- delete_node fails on random datasets
  - difficulty: finding the replacement node for the deleted node is not trivial
- The histograms I generated for insert/find random datasets were not binned correctly (to be completed by final report).
  - difficulty: (finding empirical parameters) either I should zoom in on sections of the histogram and plot that histogram or I should increase the number of insertions significantly

BU


Histogram for Time to Build Tree (example)
100
90
80
70
60
50
40
30
20
10
0
[0.00079,...
(0.01029,...
(0.01979,...
(0.02929,...
(0.03879,...
(0.04829,...
(0.05779,...
(0.06729,...


Histogram for Time for 100 find_nodes
45
40
35
30
25
20
15
10
5
0
[0.000326,...
(0.000736,...
(0.001146,...
(0.001556,...
(0.001966,...
(0.002376,...

If only I had more time . . .

- Finish implementing delete_node for random datasets
- Finish time-to-run comparisons between Quadtree and kd-tree.
- Implement nearest neighbor query and compare time-to-run with Quadtree's nearest neighbor query.

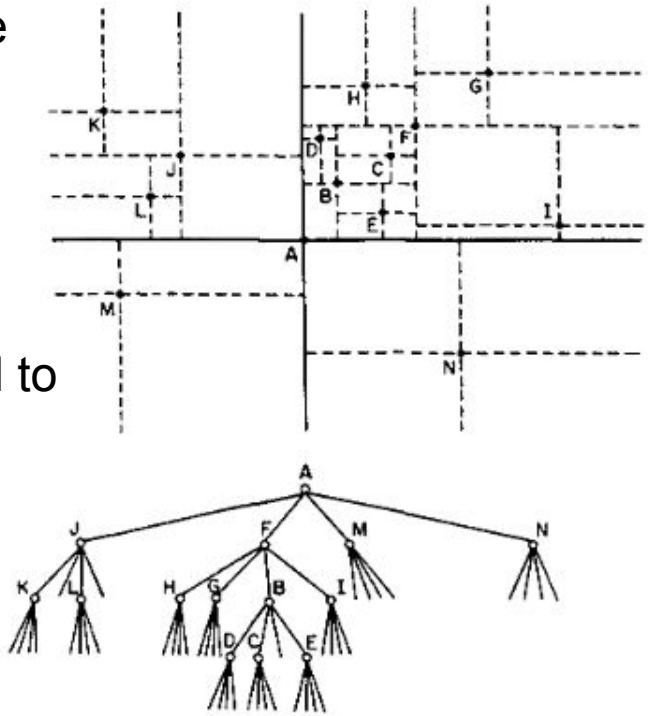
Quadtree

- Tree data structure that contains exactly four children
- used to represented 2D as a data structure
- They decompose into adaptable cells
- Have a tree pyramid but are unbalanced
- Different types:
  - Point quadtree
  - Point-region quadtree
  - edge quadtree
  - Matrix region Quadtree
- Applications
  - Image manipulation
  - Maps
  - Data compression
  - Meshing

BU

Point Quadtree

- Multidimensional generalization of BST
- Non-uniform-sized cells containing one element apiece
- Can reduce storage requirements
- Insertion
  - O(log4N)
- Deletion:
  - Average number or nodes needed to be reinserted is 67%
  - O(Nlogk+1(N))
- Search
  - O(2N^(1-1/k)

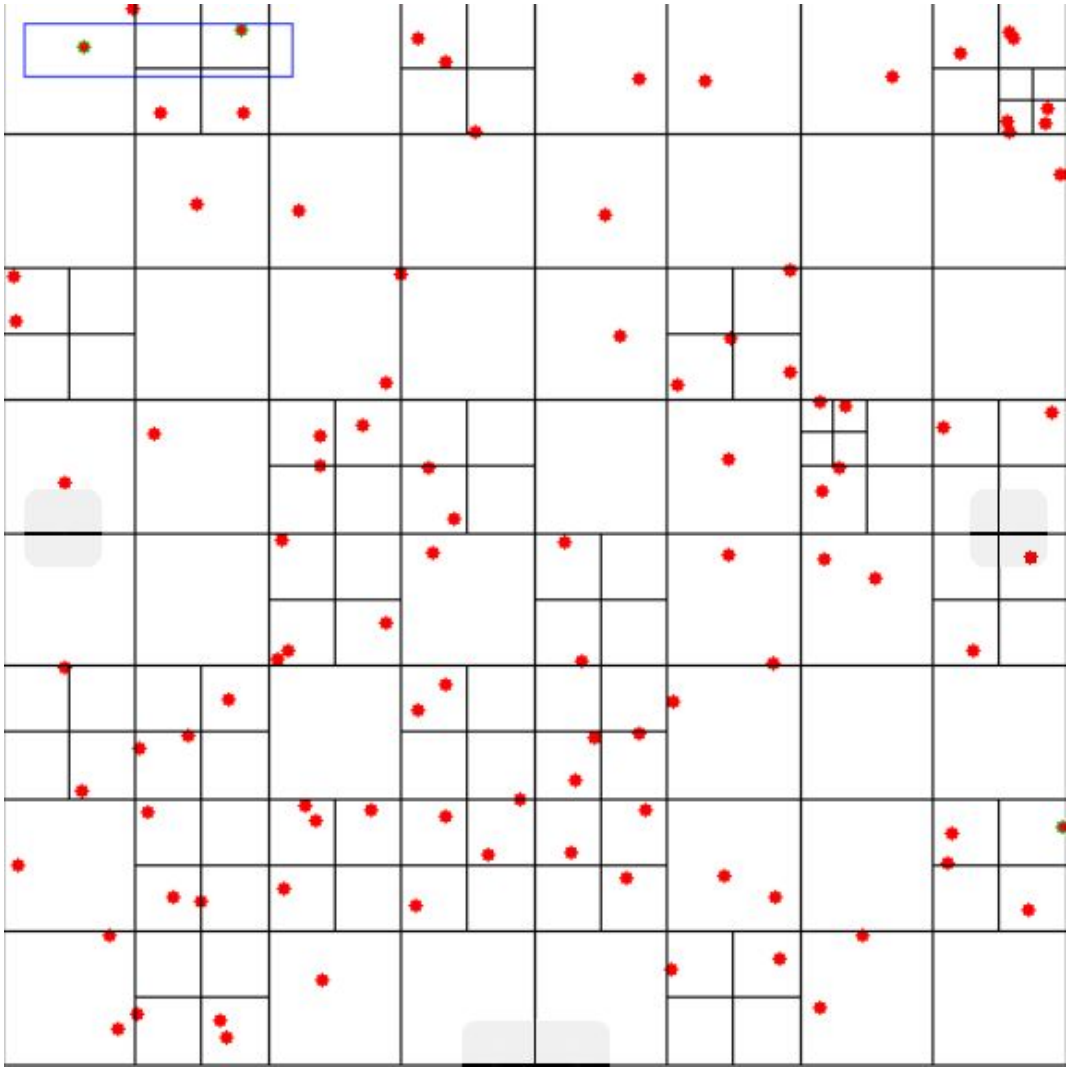


Samet, H. (1984). "The quadtree and related hierarchical data structures" (PDF). *ACM Computing Surveys*. ACM. **16** (2): 187–260. doi:10.1145/356924.356930.

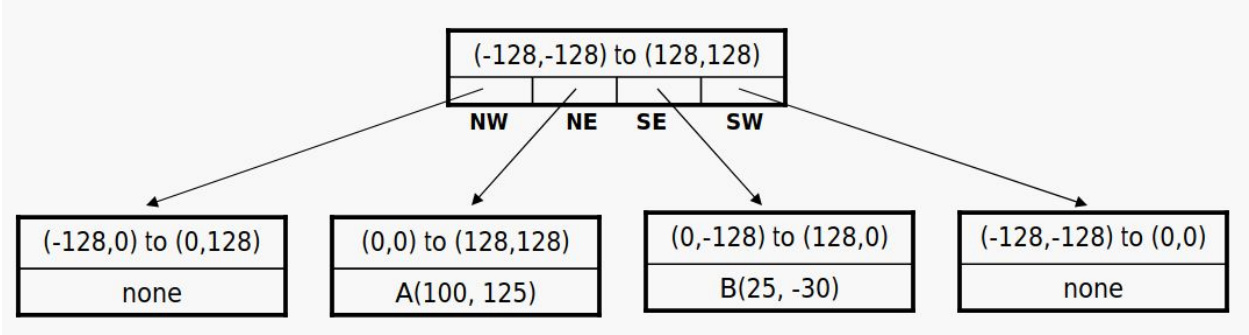
Region Based Quadtree

- Similar to Point Quadtree
- Regions are subvided equally
- Each square contains a point
- Data points are inserted into an MX quadtree by searching for them
- Insertion O(Nlog(N))

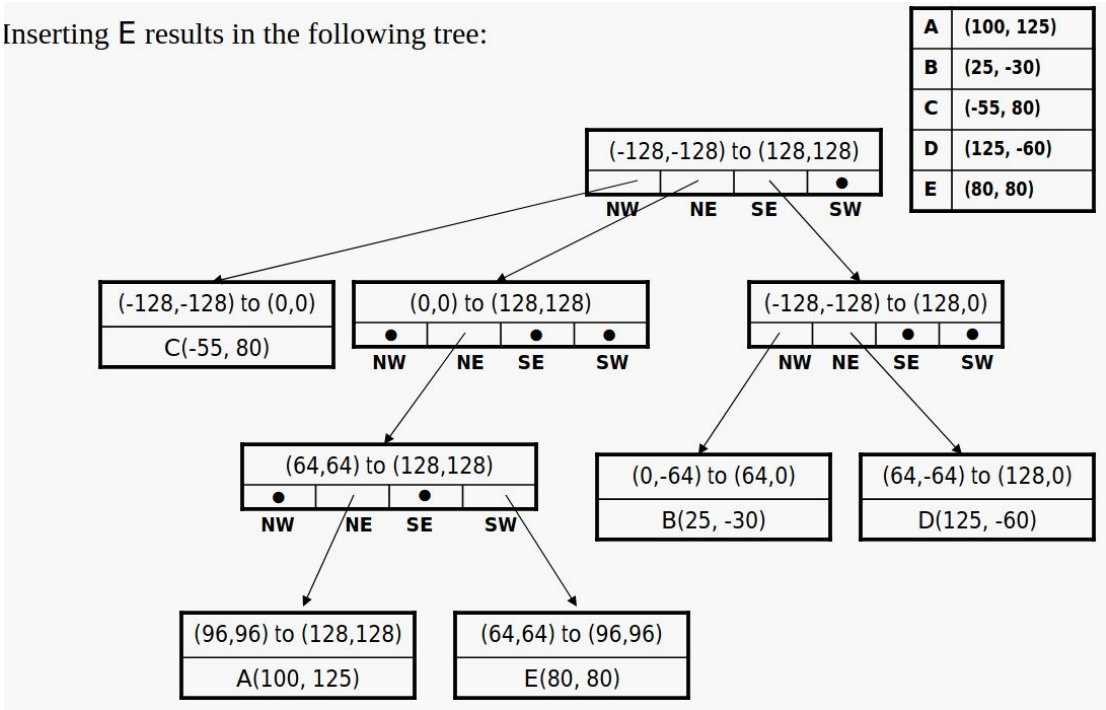
Sample implementation



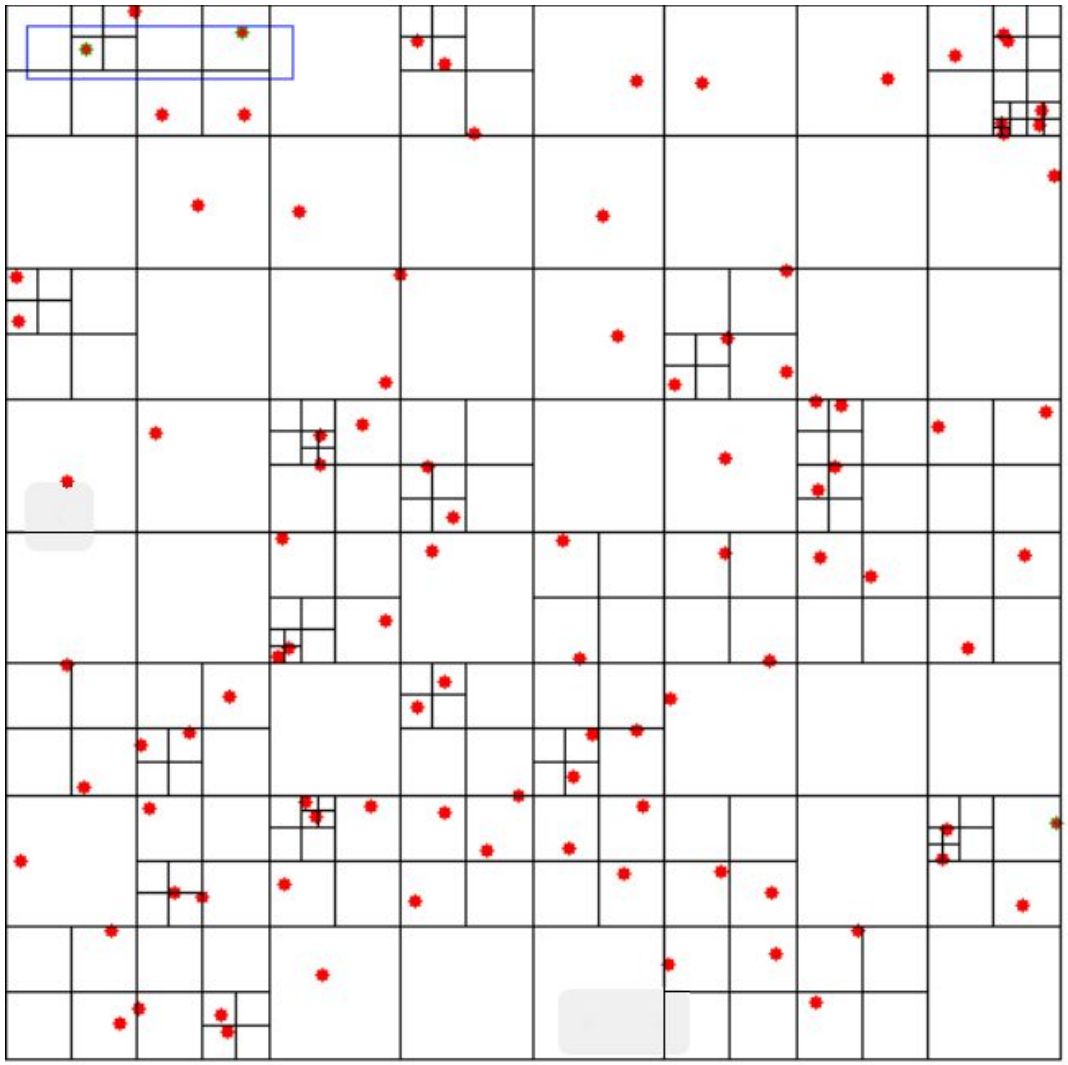
Point Region Quadtree



source:
http://courses.cs.vt.edu/~cs3114/Summer15/Notes/T04_PRQuadTrees.pdf

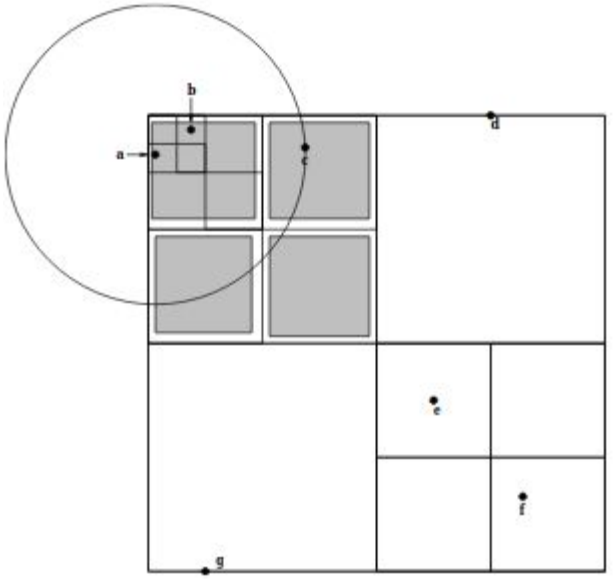


Sample Implementation

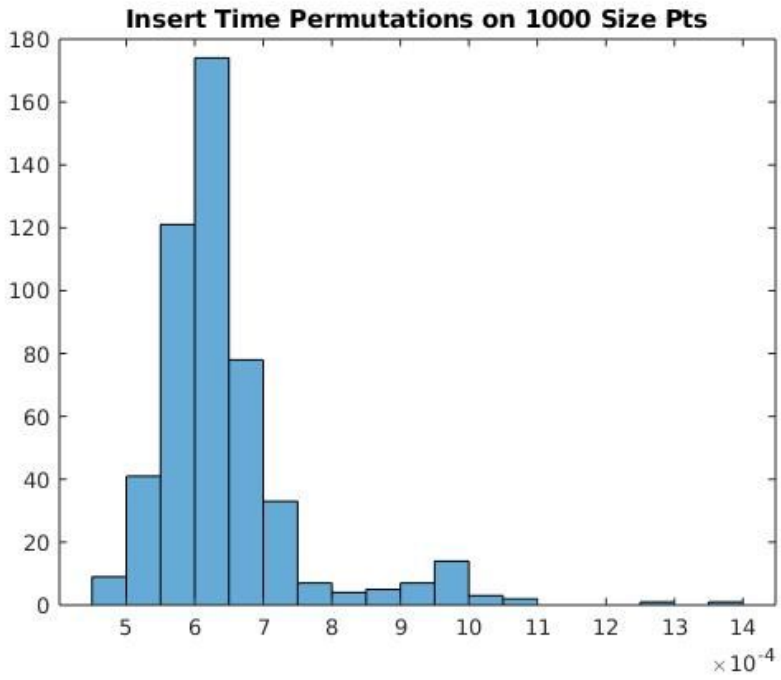


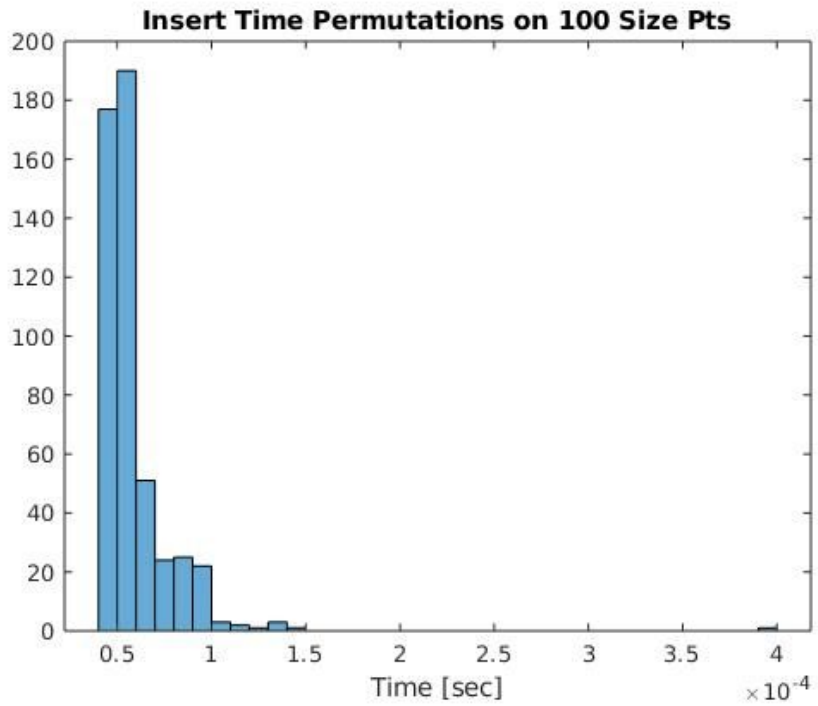
Region Find on Quadtree

- Same as Point Finding
- Also used for nearest neighbour finder
- Finds closest point in neighbourhood or region based on region



Sample Insertion Time Histogram

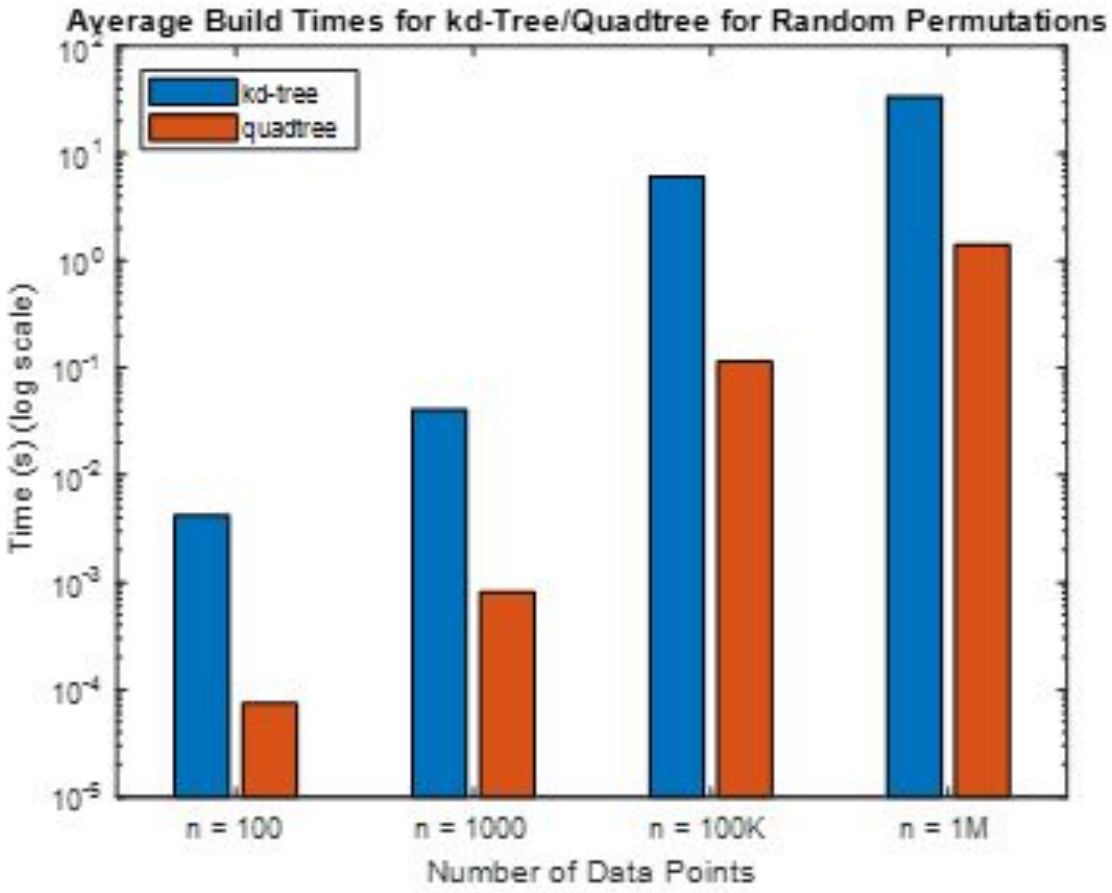




KD-TREE vs QUADTREE

AVG Time insertion
Sample 1000

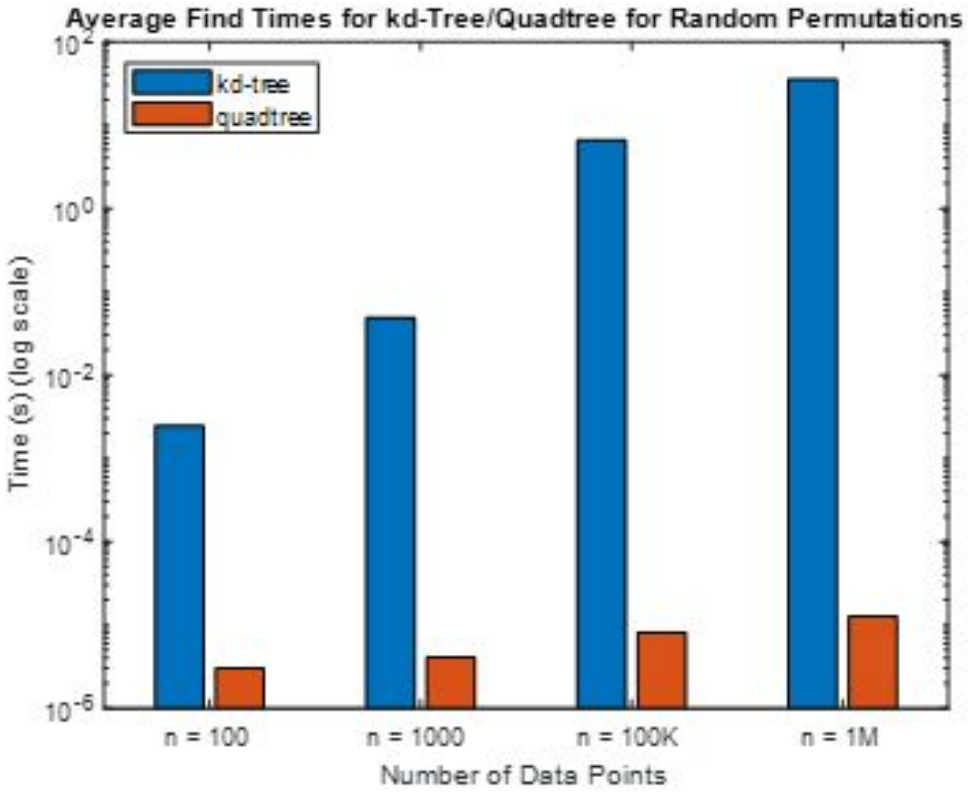
- kd-tree : 0.0054 sec
- quadtree: 0.0006 sec



KD-TREE vs QUADTREE Avg. Find Time

AVG Finding Time Sample 1000

- kd-tree: 0.0013 sec
- quadtree: 0.0014 sec



Works cited:

kd-trees: Jon Louis Bentley. 1975. Multidimensional binary search trees used for associative searching. Commun. ACM 18, 9 (September 1975), 509-517. DOI=http://dx.doi.org/10.1145/361002.361007

Quadtree:

- Samet, H. (1984). "The quadtree and related hierarchical data structures" (PDF). *ACM Computing Surveys*. ACM. **16** (2): 187–260. doi:10.1145/356924.356930.