Quadtrees

Laura Toma

Bowdoin College
Quadtree

- A data structure that corresponds to a hierarchical subdivision of the plane
- Start with a square (containing inside input data)
  - Divide into 4 equal squares (quadrants)
  - Continue subdividing each quadrant recursively
  - Subdivide a square until it satisfies a stopping condition, usually that a quadrant is “small” enough
    - for e.g. contains at most 1 point
Quadtrees

• Conceptually simple
• Generalizes to >2 dimensions
  • $d=3$: octree
• Can be built for many types of data
  • points, edges, polygons, images, etc
• Can be used for many different tasks
  • search, point location, neighbors, etc
  • dynamic
• Theoretical bounds not great, but widely used in practice
• LOTS of applications
  • Many variants of quadtrees have been proposed
  • Hundreds of papers
Point-quadtree
Point quadtree

Let $P =$ set of $n$ points in the plane

Problem: Store $P$ in a quadtree such that every square has $\leq 1$ point.

Questions:

1. Size? Height?
2. How to build it and how fast?
3. What can we do with it?
Let $P =$ set of $n$ points in the plane
Let $P = \text{set of } n \text{ points in the plane}$
Let $P = \text{set of } n \text{ points in the plane}$
Let $P = \text{set of } n \text{ points in the plane}$
Let $P =$ set of $n$ points in the plane
Let $P =$ set of $n$ points in the plane
Let $P =$ set of $n$ points in the plane
Let $P = \text{set of } n \text{ points in the plane}$
Let $P = \text{set of } n \text{ points in the plane}$

Quadtrees: trees corresponding to the subdivision
Let $P = \text{set of } n \text{ points in the plane}$

Quadtree: tree corresponding to the subdivision
Let $P$ = set of $n$ points in the plane

Quadtrees: tree corresponding to the subdivision
Let $P$ = set of $n$ points in the plane

Quadtree: tree corresponding to the subdivision
Let $P =$ set of $n$ points in the plane

Quadtrees: tree corresponding to the subdivision
Let $P = \text{set of } n \text{ points in the plane}$

Quadtree: tree corresponding to the subdivision
Let $P = \text{set of } n \text{ points in the plane}$

Quadtree: tree corresponding to the subdivision
Let $P = \text{set of } n \text{ points in the plane}$

**Quadtree**: tree corresponding to the subdivision
Let $P = \text{set of } n \text{ points in the plane}$

**Quadtree:** tree corresponding to the subdivision
Let $P = \text{set of } n \text{ points in the plane}$

Quadtrees: tree corresponding to the subdivision
Let $P = \text{set of } n \text{ points in the plane}$

Quadtree: tree corresponding to the subdivision
Let $P$ = set of $n$ points in the plane

Quadtree: tree corresponding to the subdivision
Let $P =$ set of $n$ points in the plane

Quadtree: tree corresponding to the subdivision
Exercises

- Pick n=10 points in the plane and draw their quadtree.

- Show a set of (10) points that have a balanced quadtree.

- Show a set of (10) points that have an unbalanced quadtree.

- Draw the quadtree corresponding to a regular grid
  - how many nodes does it have?
  - how many leaves? height?

- Consider a set of points with a uniform distribution. What can you say about the quadtree?

- Let’s look at sets of 2 points in the plane.
  - Sketch the smallest possible quad tree for two points in the plane.
  - Sketch the largest possible quad tree for two points in the plane.
  - An upper bound for the height of a quadtree for 2 points?

- What can you say about all points at the same level in the quadtree?
Quadtree size

P = set of n points in the plane

Theorem:
The height of a quadtree storing P is at most $\lg (s/d) + 3/2$, where $s$ is the side of the original square and $d$ is the distance between the closest pair of points in P.

Proof:

- Each level divides the side of the quadrant into two. After $i$ levels, the side of the quadrant is $s/2^i$.
- A quadrant will be split as long as the two closest points will fit inside it.
- In the worst case the closest points will fit diagonally in a quadrant and the “last” split will happen at depth $i$ such that $s \sqrt{2}/2^i = d$.
- The height of the tree is $i+1$.

- What does this mean?
  - The distance between points can be arbitrarily small, so the height of a quadtree can be arbitrarily large in the worst case.
Building a quadtree

• Let’s come up with a (recursive) algorithm to build quadtree of P

//create quadtree of P and return its root
buildQuadtree(set of points P, square S)
Building a quadtree

//create quadtree of P and return its root

buildQuadtree(set of points P, square S)
• if P has at most one point:
  • build a leaf node, store P in it, and return node
• else
  • partition S into 4 quadrants S1, S2, S3, S4 and use them to partition P into P1, P2, P3, P4
  • create a node
  • node ->child1 = buildQuadtree(P1, S1)
  • node ->child2 = buildQuadtree(P2, S2)
  • node ->child3 = buildQuadtree(P3, S3)
  • node ->child4 = buildQuadtree(P4, S4)
• return node

Let P = set of n points in the plane
Building a quadtree

//create quadtree of P and return its root
buildQuadtree(set of points P, square S)

• if P has at most one point:
  • build a leaf node, store P in it, and return node
• else
  • partition S into 4 quadrants S1, S2, S3, S4 and use them to partition P into P1, P2, P3, P4
  • create a node
  • node ->child1 = buildQuadtree(P1, S1)
  • node ->child2 = buildQuadtree(P2, S2)
  • node ->child3 = buildQuadtree(P3, S3)
  • node ->child4 = buildQuadtree(P4, S4)
  • return node

Let P = set of n points in the plane

How long does this take, function of n and height h?
Analysis

- The logic
  - Total time = total time to partition + total time in recursion

- We’ll show that
  - Partition: $O(n \times h)$
  - Recursion: $O(n \times h)$

Theorem:
A quadtree for a set $P$ of points in the plane can be built in $O(n \times h)$ time.
Let $P = \text{set of } n \text{ points in the plane}$

A quadtree for $P$ of height $h$

Partition $P$ into $P_1, P_2, P_3, P_4$ takes $O(|P|) = O(n)$

$P_1 + P_2 + P_3 + P_4 = P$

Partition $P_1, P_2, P_3, P_4$ takes $O(|P|) = O(n)$

The time to partition, at every level, is $O(n)$

$O(h \times n)$ total
Recursion

Let \( P = \) set of \( n \) points in the plane

Every recursive call creates a node

How many nodes?

- The number of nodes can be unbounded.
- We want to express \( \text{nb.nodes} \) as function of height \( h \).
Recursion

Let $P = \text{set of } n \text{ points in the plane}$

- Every recursive call creates a node
- How many nodes?
  - nodes = internal nodes + leaves
    - $N = I + L$
  - We can find a relation between $I$ and $L$
    - Each internal node has 4 children.
    - It can be shown that $L = 3I + 1$
      (proof by induction)
Recursion

Let $P =$ set of $n$ points in the plane

A quadtree for $P$ of height $h$

- Every recursive call creates a node
- How many nodes?
  - nodes = internal nodes + leaves
    \[ N = I + L \]
  - We can find a relation between $I$ and $L$
    - Each internal node has 4 children.
    - It can be shown that $L = 3I + 1$
      (proof by induction)
  - It follows that $N = I + L = 4I + 1$
Building a quadtree

A quadtree for $P$ of height $h$

- How many internal nodes?
  - Can be unbounded
  - Want to express function of $h$

- The usual argument does not work
  - each leaf contains at most one point
  - best case: no empty leaves
  - worst case: many empty leaves, many internal nodes

- At each level, each internal node contains at least 2 points
  $\Rightarrow O(n)$ internal nodes per level

$O(\ n \times h\ )\ nodes$
Summary

Theorem:
A quadtree for a set $P$ of points in the plane:

- has height $h = O(\lg (1/d))$ (where $d$ is closest distance)
- has $O(h \times n)$ nodes; and
- can be built in $O(h \times n)$ time.

- Theoretical worst case:
  - height and size are unbounded

- In practice:
  - often $h = O(n) \implies$ size $= O(n^2)$, build time is $O(n^2)$
  - For sets of points that are uniformly distributed, quadtrees have height $h = O(\lg n)$, size $O(n \lg n)$ and can be built in $O(n \lg n)$ time.
Compressed (point) quadtrees
Exercise

Let $P = \text{set of } n \text{ points in the plane}$

- Draw a quadtree of arbitrarily large size corresponding to a small set of points in the plane (pick $n=2$ or $n=3$).
  - How many leaves are empty / non-empty?
  - Why is the size of the quadtree super-linear?

- Compress the quadtree as follows:
  - Compress paths of nodes with 3 empty children into one node
  - This node is called a \textit{donut}
  - A node may have 5 children, an empty \textit{donut} + 4 regular quadrants
Compressed quadtrees

Let $P =$ set of $n$ points in the plane

- A compressed quadtree is a regular quadtree where paths of nodes with 3 empty children are compressed into one node (called: donut)
- A node may have 5 children, an empty *donut* + 4 regular quadrants
Compressed quadtrees

• A compressed quadtree is a regular quadtree where paths of nodes with 3 empty children are compressed into one node (called: donut)

• A node may have 5 children, an empty *donut* + 4 regular quadrants

• What does this mean in terms of size?

Theorem:  A compressed quadtree has $O(n)$ nodes and $h=O(n)$ height.

• Proof idea: For each leaf that’s empty and for each donut, there exists one sibling leaf that’s not empty. The number of non-empty leaves is $n$. 
Applications of quadtrees

- Hundreds of papers
- Specialized quadtrees
  - customized for specific types of data (images, edges, polygons)
  - customized for specific applications
  - customized for large data
- Used to answer queries on spatial data such as:
  - point location
  - nearest neighbor (NN)
  - k-NNs
  - range searching
  - find all segments intersecting a given segment
  - meshing
  - ...