Minute essay comment:
- Is Scala better than other languages at context-sensitive grammars?
- IcP solutions.
Lots of applications have data that has a spatial component to it. Simple examples are characters in a world or particles in space.

Consider you want to find all other things within a certain distance of one of them.

How do you do this? What is the order of your approach?
One way to speed this up is to just put everything into a grid.

Make a 2-D array of linked lists. Make grid cells the size of the region you are searching. Only have to look in adjacent cells.

This works fine if the distribution is fairly uniform. When the distribution isn't uniform it has problems.

It also can run into memory issues above 2-D.
We saw earlier that trees are highly flexible. The BST can do some things like an array, but perform all operations in $O(\log n)$ time.

We can also make trees that break things up spatially.

These deal well with non-uniform distributions and also give roughly $O(\log n)$ performance for single operations.

You can use recursion to search for neighbors.
As the name implies, this type of tree has four children at each node.

Typical implementation starts with a box and recursively splits it in the center along x and y.

Data goes in leaves and splitting stops when you get down to a certain number of particles.

Octree is the equivalent in 3-D. Approach doesn't scale well above that.
This is a move flexible option that splits on a particular axis at each node. (e.g. x=3)

Internal nodes have two children.

Split direction can change regularly or go in the direction of greatest spread.

Scales well to high dimensions. Can be built in a way that is perfectly balanced.

Let's write one.
Other Options

- There are lots of options for these.
- Point based trees keep data in all nodes and split on points.
- BSP-trees used in games/graphics. (Binary Space Partition) Like kD-trees, but split can be an arbitrary plane.
- etc.
If we were going to spend more time on any one topic we have covered, what would you want it to be?