kbrown24
k-d trees
final project proposal

To my understanding k-d trees are data structures that partition k-dimensional data sets, so they can be navigated, searched on, and worked with more reasonably. They are often used for nearest neighbor searches in k-dimensional data sets.There are many ways to partition the k-dimensional data sets, but it seems that one common way is to choose a random dimension/attribute and to 'split' on that attribute by finding the median value for that attribute in the data and splitting the data into lower-dimensional data sets, following this process recursively until some desired number of data points resides within each region or the data has been split by some number of the k-dimensions. Because I haven't taken cs0320 and I'm still learning about k-d trees from different resources, my understanding of them is pretty rudimentary.

I chose this topic because I really like data structures in general. The idea of multidimensional trees seems really interesting to me, and I'm excited to get really familiar with k-d trees over the duration of the project. I also like the way k-d trees can look geometrically. Also, my partner, pharvie, has taken cs0320, so I am happy for the clarity he can bring to the topic as I learn about k-d trees. Mostly I just like data structures though.

foundation goal: correctly modeling/visualizing 1 to 2 dimensional k-d trees and finding instances in Alloy that prove why a more complicated nearest neighbor algorithm is necessary for complete nearest neighbor accuracy when using k-d trees.

target goal: model the nearest-neighbor algorithm itself and show it is correct within reasonably small bounds.

reach goal: model the more complex nearest-neighbor algorithm for always finding the true nearest neighbor; model k-d trees of 3 dimensions or higher.

Because I'm not entirely sure how hard it will be to model k-d trees of low dimension, I'm not sure if these goals aren't ambitious enough, but better safe than sorry with Alloy.