**K-D Tree With Core Location and MapKit in Swift**

Part 1: Intro to KD Trees and Demonstration of Concepts

InstaData

With the rise of big data, modern applications are expected to work with large data sets and react in small time frames. Demand for the retrieval of information instantly is high and can be the difference between an award winning app and a 2-star dud. Although processor speed is at an all time high, data sets continue to grow and computational devices continue to shrink. A computer scientist must consider these concepts when designing an algorithm to handle such data sets on such devices and adjust accordingly.

K-D Tree Introduction

Multiple nearest neighbor and query search algorithms exist in the computer science world today. These algorithms implement different variations of tree data structures to obtain near constant nearest neighbor search and sub constant query runtimes.

Concept

A KD Tree works by using of a few simple concepts and properties. Firstly, different depths of the tree represent different axis. Secondly, a median point, a median value, or a mean of the “current” list is determined and used as an influencer. Lastly, leaf nodes are represented as actual points from the data set.

Alternating Axis

Influencer

Query and Nearest Neighbor