PFP Project Proposal

Team: Ge Wang (gw2372) and Samurdha Jayasinghe (sj2564)

Problem: Searching through a set of boundaries to find which subset contains a query point.

Input Data: For this project we will use two open source datasets, one containing polygons outlining boundaries for all countries in the world, and another dataset which contains polygons outlining states.

Solution:

The brute force approach to solving this problem involves using the ray casting algorithm (https://en.wikipedia.org/wiki/Point_in_polygon#Ray_casting_algorithm) on the entire set of polygons. This approach is both slow and uninteresting.

We will instead implement an R-tree data structure which indexes polygons based on containment, where the node at the root of a subtree spatially contains nodes below it. The search algorithm will recursively test for containment starting from top to bottom of the tree.

At each node, it will first perform an approximate bounding-box containment test to quickly check whether a point definitely does not fall within a polygon. If the bounding-box test passes, it will perform the more expensive ray casting method to determine that the point indeed falls within the precise boundary of the polygon.

We will attempt to complete this project in the following stages:

Stage A: Non-parallel implementation without any optimizations.
Stage B: Parallel implementation without additional optimizations.
Stage C: Parallel implementation with additional optimizations.

Stage D (stretch goal): Implement the approach used by Google (http://blog.christianperone.com/2015/08/googles-s2-geometry-on-the-sphere-cells-and-hilbert-curve/)

Time permitting, we will also attempt the following in order to develop familiarity with other Haskell libraries:

- Querying over a REST endpoint.
- Adding custom polygons over a REST interface, with data persisted to postgres.

Inspired By:

https://eng.uber.com/go-geofence/
https://medium.com/@buckhx/unwinding-uber-s-most-efficient-service-406413c5871d