HW2: 20 points

In this homework, you are going to work with **spatial data** - you will create some data, visualize it, do queries on it, and visualize the query results.. Hope you have fun with this!

The exercise will give you a taste of working with spatial data, use of a spatial file format and spatial query functions, all of which are quite useful from a real-world (or job interview) perspective.

What you need to do is described below in sufficient, but not too much, detail - you'd need to do a bit of reading up and experimenting, to fill in the gaps. Please talk to a TA/grader if you are unable to proceed at any point!

1. You need to create (generate) latitude,longitude spatial coordinates for 9 locations. One of those will be where your home/apartment/dorm room is. The other eight would have to be spread out - spatially distinct, at least 100 feet between adjacent locations (and at most 200 feet - we don't want to cover a huge region!). If you are on campus, you can obtain the coords of the four corners (Exposition/Vermont, Vermont/Jefferson, Jefferson/Figueroa, Figueroa/Exposition), and get coordinates for four spots inside the campus (classrooms, labs, offices, restaurants, landmarks..). If you are a DEN student, get your 8 coordinates from your place of work or neighborhood (again, make sure they are not too close to each other or too far apart).

How would you obtain spatial coordinates at a location? You can do so one of two ways:

- **using the Chrome browser**, simply bring up this (geolocate_mod/geolocate_mod.html) page on your smartphone (that has GPS), and write down the (latitude,longitude) values that get shown when you load/refresh the page :) As you can see, the page shows your location on a map - cool! Be sure to enable cross-site script loading when you run this (because the script is on our Dropbox area, and accesses a map API at google.com) - click on the shield icon at the right of the URL bar, and click on 'Load unsafe scripts'. Alternately, you can use this (geoloc2/run.html) page to obtain the (latitude,longitude) coordinates.
- using your phone's built-in GPS/compass app, simply read off the displayed GPS coordinate values (if the coordinate display is in degrees, minutes and seconds, you need to convert the minutes,seconds pair of values into a single fractional degree value - one degree is subdivided into 60 minutes (60'), and one minute is subdivided into 60 seconds (60'') - so for example, 30'15", since it is equivalent to 1815", would be eqvt to 1815/3600=0.504 degrees.

Also, be sure to write down the location names as well (you will use them to label your points when displaying).

2. Now that you have 9 coordinates and their label strings (ie. text descriptions such as "Tommy Trojan", "SAL", "Chipotle"..), you are going to create a KML file (.kml format) out of them using a text editor. Specifically, each location will be a 'placemark' in your .kml file (with a label, and coords). Here (https://developers.google.com/kml/documentation/kml_tut#placemarks) is more detail. The .kml file with the 9 placemarks is going to be your starter file, for doing visualizations and queries. Here (data/starter.kml) is a .kml skeleton to get you started (just download, rename and edit it to put in your coords and labels). NOTE - keep your labels to be 15 characters or less (including spaces).

3. Download Google Earth (https://www.google.com/earth/download/ge/agree.html) on your laptop, install it, bring it up. Load your .kml file into it - that should show you your 9 sampled locations, on Google Earth's globe :) Take a snapshot (screengrab) of this, for submitting.

4. Install Oracle 11g+Oracle Spatial, or Postgres+PostGIS on your laptop, and browse the docs for the spatial functions. You will use the software to execute the following two spatial queries that you'll write:

* compute the convex hull for your 9 points [a convex hull (http://mathworld.wolfram.com/ConvexHull.html) for a set of 2D points is the smallest convex polygon that contains the point set]. If you use Oracle, see this (https://docs.oracle.com/cd/A97630_01/appdev.920/a96630/sdo_aggr.htm) page; if you decide to use Postgres, read this (http://postgis.net/docs/ST_ConvexHull.html) and this (http://stackoverflow.com/questions/10461179/k-nearest-neighbor-query-in-postgis) instead. Use the query's result polygon's coords, to create a polygon in your .kml file (edit the .kml file, add relevant XML to

specify the KML polygon's coords). Load this into Google Earth, visually verify that your 9 points are inside the convex hull, then take a screenshot.

* compute the three nearest neighbors of your home/apt/dormroom location [look up the spatial function to do this]. Use the query's results, to create three line segments in your .kml file: line(home,neighbor1), line(home,neighbor2), line(home,neighbor3). Verify this looks correct using Google Earth, take a snapshot.

Note - it *is* OK to hardcode points, in the above queries! Or, you can create and use a table.

Here is what you need to submit (as a single .zip file):

* your .kml file from step 4 above - with the placemarks, convex hull and three nearest-neighbor line segments (2 points)

* a text file (.txt or .sql) with your two queries from step 4 - table creation commands (if you use Postgres and directly specify points in your queries, you won't have table creation commands, in which case you wouldn't need to worry about this part), and the queries themselves (12 points)

* screengrabs from steps 3,4 (3+3 = 6 points)

HAVE FUN! From here on out you know how to create graphics (KML files containing vector symbols constructed from points, lines and polygons) overlaid on any map :)
