Introduction to Spatial Computing- CSE 5ISC

Homework 1

Due date: August 18, 2015 6:00pm

Instructions:

- All submissions must be made through usebackpack site for this course (https://www.usebackpack.com/iiitd/m2016/cse5isc)
- Only one submission per team would be considered and graded. It would be assumed that all members of the team have participated equally and same score would be given to all members of the team.
- Your submission should have names of all the members of your team.
- Any assumptions made while solving the problem should be clearly stated in the solution.
- Question 3 is for teams of size 3. This questions will not be graded for teams for size 2 or less.

Question 1 (30 points) (Programming assignment on Geo-coding): For this question you are expected to use the Google Maps Javascript API. Please browse through their help site (https://developers.google.com/maps/documentation/javascript/examples/) and guide https://developers.google.com/maps/documentation/javascript/tutorial to familiarize yourself. After that, following tasks are to be performed.

(a) Select any two stations from the Delhi Metrorail map (http://www.delhimetrorail.com/Zoom_Map.aspx) which are reasonably far apart (about 5-6 stations or more). Write a javascript+html code to visualize the route taken by delhi metro using the polyline API of google (https://developers.google.com/maps/documentation/javascript/examples/polyline-simple). Note that for this you would need get the latitude and longitudes of intermediate stations. You can do so using a combination of marker API (https://developers.google.com/maps/documentation/javascript/examples/marker-simple) and click events (http://stackoverflow.com/questions/6374329/get-latitude-and-longitude-of-marker-onclick).

(b) Review the directions API of Google Maps Javascript API (https://developers.google.com/maps/documentation/javascript/directions?hl=en). Write a javascript+html code to visualize the driving route returned by google maps between any two points in the maps. Input should be taken in form of clicks, i.e., one click event for the source and another click event should be for the destination.

Question 2 (60 points) (Programming assignment on spatial data):

In this question you would be evaluating some traditional data structures for a typical spatial query. To get started, first, create the following three synthetic datasets. Here, each data point is a (X,Y) coordinate. In each of these datasets, the points are distributed slightly differently, e.g., dense (Dataset A), sparse (Dataset C), or skewed (Dataset B).

Datasets for evaluations:

Following three synthetic datasets should be created and used for evaluation:

Dataset A: 50000 points generated in a uniformly random fashion where the x and y coordinates are integers between 0 and 900.

Dataset B*:* 25000 points generated in a uniformly random fashion where x and y coordinates are integers between 0 and 600. And another 25000 points generated over the ranges of 600 – 1600 (integer x and y coordinates)

Dataset C: 50000 points generated in a uniformly random fashion where x and y coordinates are integers between 0 and 1200.

Candidate Data Structures: Implement the following two data structures for the spatial data.

(a) **1-Dimensional Hashing:** Here, we would create a key by combining both the X and Y coordinates. Your key could be an n-bit integer with first n/2 bits as the x-coordinate and the last n/2 bits as the y-coordinate. Hash this key using any hash function with chaining for handling collisions. In order to simulate secondary memory constraints, your code should support the notion of different bucket sizes (to be varied in experiments). Each unique value of hash-function should be a different bucket. Once a bucket overflows, you need to create a new bucket and link it to current one.

(b) **2-Dimensional Grid:** Here, we would create a 2-dimensional grid to store the data points. Similar to the 1-dimensional hashing, each grid cell should be stored as a bucket. For example, if your grid cell size is 100-by-100, all the points whose x-coordinate is between 0 – 100 and y-coordinate is between 0 – 100 would go into one bucket. Once a bucket overflows, you need to create a new bucket and link it to current one (similar to chaining).

Spatial Query Processing Algorithm:

Develop a k-nearest neighbor query algorithm for the 1-Dimensional Hashing and 2-Dimensional Grid data structure. A k-nearest neighbor query algorithm takes a value of *k* and a query point (X,Y) as an input. The output contains *k* data points from the dataset which are closest to the given query point than any other point in the dataset.

Experiments to be conducted:

Experiment 1:

Dataset to be used: Dataset A

Algorithms to be compared: 1-Dimensional Hashing and 2-Dimensional Grid (cell size 100-by-100).

Variable Parameters: Bucket sizes 10, 40, 70, 100, and 150

Metric to be measured: Average Run-time in seconds for 10 random k-nearest neighbor queries.

Plot a graph of the data collected in the experiment for k=5 and k=10

Experiment 2:

Dataset to be used: Dataset A

Algorithms to be compared: 1-Dimensional Hashing and 2-Dimensional Grid (cell size 100-by-100).

Variable Parameters: #points inserted so far into the data structure. Take reading after every 1000 points.

Metric to be measured: Space utilization (=total #data points/total capacity of buckets)

Plot a graph of the data collected in the experiment for bucket sizes 10 and 70.

Experiment 3:

Repeat Experiment 1 with Dataset B

Experiment 4:

Repeat Experiment 2 with Dataset B

Experiment 5:

Repeat Experiment 1 with Dataset C

Experiment 6:

Repeat Experiment 2 with Dataset C

Question 3 (20 points): Answer the questions (a) through (f) from Question 3 in the context EER diagram shown below. Here the answers are restricted to 0, 1 or many followed by a very brief justification of the answer. Does this EER diagram also violate any of your knowledge of geometry? Give an example "area" which is accepted by the following EER but is not a valid area from a geometric sense. Limit your narrative to 100 words.

(a) What is the maximum number of directed arcs associated with an area?
(b) What is the minimum number of directed arcs associated with an area?
(c) What is the maximum number of left-bounding area for a given arc?
(d) What is the minimum number of left bounding area for a given arc?
(e) What is the maximum number of nodes for a given area?
(f) What is the minimum number of nodes for a given area?

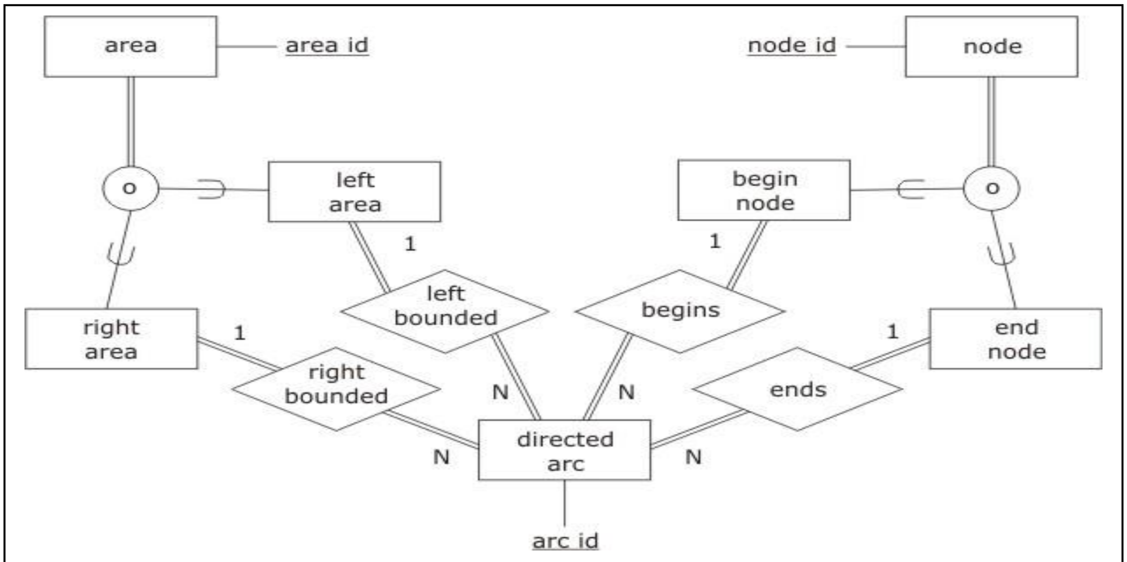

Figure for Question 3.

Things to be submitted:

A zipped folder containing the following items. Please include your team information with the submission.

(a) Code and screen shots for Question 1.
(b) Code, query processing algorithm, plots of Question 2. Also include a brief analysis of the trends obtained in the experiments (about 400 words).
(c) A pdf file containing solution for Question 3.