Introduction to Spatial Computing- CSE 5ISC

Homework 3

Due date: Nov 8, 2015 23:59pm

Instructions:

- All submissions must be made through usebackpack site for this course (www.usebackpack.com/iiitd/m2015/cse5isc/info)
- Only one submission per team would be considered and graded. It would be assumed that all members of the team participated equally and same score would be given to all members of the team.

Question: Programming Based

In this question you would be comparing the performance of some spatial indexing techniques for range queries on some spatial points.

Index structures need to be implemented (60 points):

(a) Z-order curves. You are strongly suggested to use existing code for B+ tree. In other words, just code the z-order values and insert them into a B+ tree created from an existing code found online.
(b) Hilbert curves. Similar to part (a), you are strongly suggested to use existing code for B+ tree.
(c) Grid files.

Datasets for evaluations:

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

Dataset A: 1000 points generated in a uniformly random fashion where the x and y coordinates are integers between 0 and 500.

Dataset B: 500 points generated in a uniformly random fashion where x and y coordinates are integers between 300 and 500. And another 500 points generated over the ranges of 0 – 300 (integer x and y coordinates)

Dataset C: 700 points generated in a uniformly random fashion where x and y coordinates are integers between 300 and 500. And another 300 points generated over the ranges of 0 – 300 (integer x and y coordinates)

Bucket sizes:

Option-one: 5 points *Option-two:* 20 points

Query Algorithm to be implemented on index structures (20 points)

K-nearest neighbor (Knn) query: Given the coordinates of a query point retrieve k points which are closest to it than any other points in the datasets. Your answer should include a description of the Knn algorithm for each of the index structures and a brief note on why it is correct.

Analysis to be done (20 points)

Randomly generate Knn queries and determine the number of buckets examined to retrieve the final results. Following plots should be submitted with the answer. All plots should have proper legends.

Plot 1a containing 3 curves:

X-axis: Value of k. Pick k=1, 5, 10 and 15

Y-axis: Average number of buckets retrieved for 10 randomly generated Knn queries

Bucket size: 5 points

Curve 1: Z-order curves on Dataset A

Curve 2: Hilbert curves on Dataset A

Curve 3: Grid files on Dataset A

Plot 2a containing 3 curves:

X-axis: Value of k. Pick k=1, 5, 10 and 15

Y-axis: Average number of buckets retrieved for 10 randomly generated Knn queries

Bucket size: 5 points

Curve 1: Z-order curves on Dataset B

Curve 2: Hilbert curves on Dataset B

Curve 3: Grid files on Dataset B

Plot 3a containing 3 curves:

X-axis: Value of k. Pick k=1, 5, 10 and 15

Y-axis: Average number of buckets retrieved for 10 randomly generated Knn queries

Bucket size: 5 points

Curve 1: Z-order curves on Dataset C

Curve 2: Hilbert curves on Dataset C

Curve 3: Grid files on Dataset C

Plot 1b: repeat the experiment in Plot 1a with a bucket size of 20 points

Plot 2b: repeat the experiment in Plot 2a with a bucket size of 20 points

Plot 3b: repeat the experiment in Plot 3a with a bucket size of 20 points

Things to be submitted:

A zipped folder containing the following:

(a) Code for Z-order, Hilbert and Grid files
(b) Query algorithm for KNN search
(c) The above mentioned six plots