INFS7205
Assignment 2

Research Proposal

Group 1

Sarah Murray	45288069
Yohan MIchaux	45021426
Yuhua Pu	43120110
Hanning Gu	45207441
Xiaochang Xu	44880345

Table of Contents

1. Introduction

Our group will focus on Type B implementation of a Spatial Database Management System (SDBMS) to test the performance of Point queries on data indexed by a quad-tree.

The SDBMS will support standard non-spatial attributes, as well as the spatial data types Point and Region. Point will represent the longitude and latitude of a position in geographical space, and Region will represent the boundary of an area.

The tables in the SDBMS will be indexed using a quad-tree on this Point attribute. The performance of using a quad-tree will be analysed based on the speed and memory usage of a number of Point queries.

To test the performance, the queries will be carried out firstly on the data when indexed by the quad-tree, then secondly when no indexing structure is used.

The aim of the project is to test the performance of the quadtree in terms of spatial queries, it does not take into consideration the creation, maintenance or storage of the tree.

2. Proposal

The same spatial dataset will be used in all of the following proposed implementation for comparison purpose.

2.1 Basic Database Operations

Our code will support standard relational database operations which will allow users to insert/update/delete spatial objects (POINT/POLYGON), then it will recursively subdivide the quadrant until the specified threshold or resolution is met.

2.2 Point/Window query

In this part, we will mainly realize the three types of queries in Assignment 1 with simulated spatial dataset.

The indexed dataset will be able to return all of the objects (either a point or a polygon) that intersect with the given query point/window.

2.3 K-Nearest Neighbour

We will also test if our code would return the K-Nearest Neighbour(s)

2.4 Performance Evaluation

Finally, we will compare the memory and time efficiency with the unindexed dataset where all of the above operations are done by brute-force. This is also done in Python.

3. Allocation

3.1 Dataset

Obtain spatial datasets from various source and preprocess them to suit our needs in implementation phase (1-2 people)

3.2 Implementation

Use Python Quadtree package to index the dataset and then test on point query, window query and KNN which corresponds to A1 Q2).

The memory usage and time efficiency are then compared against unindexed dataset (2-3 people)

4. Design Choices

We will use Python on Windows/MacOS to build the code for our implementations.

The dataset we'll be using include real-world spatial data as well as random generated data for test purposes.