Projects: Implementation and Experimental Evaluation of Multidimensional Data Structures

Professors: S. Sioutas, K. Tsichlas, G. Vonitsanos (Postdoc Researcher@CEID)

Goal: The major task is the implementation and experimental evaluation of a variety of multi-dimensional data structures in a programming language of your preference (we suggest Python, C++ or Java). You could use artificial synthetic-data sets or real-data sets to evaluate the performance of the following fundamental operations: Build, Insert, Delete, Update, Searching (Similarity, kNN) Queries.

You can download real datasets from the following URLs:

Find Open Datasets and Machine Learning Projects | Kaggle

20 Free Datasets for Data Science Projects | Built In

https://freegisdata.rtwilson.com/

https://paperswithcode.com/datasets?task=trajectory-forecasting

https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html

*****Choose one of the following two (2) projects:**

Project-1: **Multi-dimensional Data Indexing and Similarity Query Processing**: Develop Multidimensional Access Methods based on **k-d trees, quad trees, Range Trees and R-trees** respectively to support ***k-dimensional queries***. Consider the case where k<=4 (indexing on 4 at most attributes). Then, after the 1st phase of indexing, in a 2nd phase, perform ***similarity queries*** according to a specific textual attribute (f.e. review comments, etc) based on **LSH** technique. The final task is an exhausted evaluation performance comparison among the 4 proposed schemes: **k-d + LSH, Quad + LSH, Range + LSH, R-trees + LSH**.

Example: Consider the Coffee Reviews Dataset (Coffee Reviews Dataset) from Kaggle. This dataset organizes global reviews of coffee between 2017 and 2022 based on factors like blend name, type of roast, price and geographical origin of coffee beans. It is pre-processed and cleaned, and can be used for pandas, data engineering, analysis and feature engineering practice. The original version of the dataset comes with 12 features, while the simplified version has 9 features. *f.e. we would like to detect the N-top most similar Reviews (documents) conducted during 2019 up to 2021, took review-rating more than 94, it's price per 100g (100g_USD) is between 4$ and 10$, and the country origin (loc country) is USA, where N is a user defined parameter (f.e. N=3).*

Project-2: Develop the following geometric data structures. We suggest the following real data set:

- https://paperswithcode.com/datasets?task=trajectory-forecasting
- https://freegisdata.rtwilson.com/

1. **3D R-trees for Spatio-Temporal Query Processing in a dataset of planar trajectories**: Implement 3-dimensional R-trees to index trajectories of moving object on the plane. Each trajectory is a set of 3-dimensional points of the format (x,y,t), representing the spatial position (x,y) of mobile object at time

instant t. Evaluate experimentally the time performance of 3-dimensional range queries. For example, queries that select moving objects passed from a specific spatial terrain during a specific time interval [t1, t2]. f.e. "Find the number of vehicles passed from Olgas' Square with spatial coordinates [x1, x2] x [y1, y2] from 12:00 am up to 14:00 am".

2. **Interval trees** and **Segment trees**. Evaluate the time performance of the basic operations, ***interval and stabbing Queries*** respectively and prove experimentally that time responses follow the theoretical complexity.

3. **Convex Hull:** Develop CH algorithms for 2 and 3 dimensions. Evaluate the time and space performance of your proposed method and prove experimentally that the response time follows the theoretical complexity.

4. **Line Segment Intersection:** Develop Line Segment Intersection Algorithms using the basic sweep line technique. Evaluate the time and space performance of your proposed method and prove experimentally that the response time follows the theoretical complexity.

Background Knowledge: Data Structures, Algorithms and Complexity, Databases, Object Oriented Programming (C++, JAVA), Functional Programming (Python, Scala).

References:

1. Book ("advanced data structures", A.K. Tsakalidis)
2. https://en.wikipedia.org/wiki/Range_tree
3. https://en.wikipedia.org/wiki/K-d_tree
4. https://en.wikipedia.org/wiki/Quadtree
5. https://en.wikipedia.org/wiki/Interval_tree
6. https://en.wikipedia.org/wiki/Segment_tree
7. https://en.wikipedia.org/wiki/Priority_search_tree
8. https://en.wikipedia.org/wiki/Bloom_filter
9. https://en.wikipedia.org/wiki/MinHash
10. https://en.wikipedia.org/wiki/R-tree
11. https://en.wikipedia.org/wiki/Convex_hull
12. https://en.wikipedia.org/wiki/Voronoi_diagram
13. https://en.wikipedia.org/wiki/Sweep_line_algorithm
14. https://en.wikipedia.org/wiki/Line_segment_intersection

(***) Deliverables: Zip or Rar file with executable files. Deadline: ~ 1st WEEK of February, 2024.