Department of Computer Science and Engineering
The Hong Kong University of Science and Technology

CSIT6000P Spatial and Multimedia Databases

2022 Spring

Version 1.0

Assignment 1 [*Total: 30 marks*]

Due date: 11:59pm Saturday 26 March 2022
HKUST Canvas online submission only.

You are given a simplified real-world database *D* of points-of-interest (POIs), such as restaurants, schools, shops, and bus stops. After removing all sensitive information, each record contains only an ID and a location represented as (*x*, *y*) for its longitude and latitude values. In this assignment, you are asked to conduct an experimental study on the performance of spatial indexing methods on the given dataset and report your findings. This assignment only considers a simplified scenario where all the data are already loaded into memory. That is, no disk-based operations need to be considered.

Task 1 [*6 mark*] (**grid indexing with regular decomposition**)

(1) [*1 mark*] Write a program to compute the MBR for the POI dataset *D*.
(2) [*2 mark*] Experimentally find a resolution *n* such that the maximum number of points in any of the 2^n x 2^n regularly decomposed grid cells will not exceed 128.
(3) [*1 mark*] Report the number of cells with no more than 5 points based on grid decomposition above.
(4) [*2 mark*] Design and create an index using the 2^n x 2^n grid cells obtained above (i.e., each grid cell has a pointer to the *D* points in the grid cell). An index is defined in the form of a list of (*r*, *b*) pairs, where *r* is a rectangle and *b* is a pointer to a bucket that holds all points in *D* inside *r*. Rectangle *r* is called the index cell. The capacity of each bucket is fixed at 128 (that is, a bucket can hold no more than 128 points) – the capacity constraint has already been considered in subtask 2 above. Empty grid cells do not need to be recorded.

Task 2 [*9 marks*] (**z-value indexing with adaptive decomposition**)

(1) [*7 mark*] For the resolution number *n* above, write a program to generate the base-5 z-value for each point in *D*. The decomposition terminates either when reaching the resolution level, or there are no more than 5 points in a Peano cell. The entire space is represented by 1, so all z-values must start from 1. For the pints in any early-termination cells, 0s should be appended at the end of their z-values to make all z-values equal length. The marks for this subtask will be given by the assessor based on if your program is correct and properly commented.
(2) [*2 marks*] Design and create an index by sorting the z-values of all point data in *D* (i.e., all z-values are sorted, and each z-value is associated with a point in *D* with that z-value). Note that it is possible for many points to share the same z-value.

Task 3 [*11 marks*] (**window query processing**) A window query with a given query rectangle represented as $Q = \{(x_{low}, y_{low}), (x_{high}, y_{high})\}$, returns the number of points inside *Q* (i.e., $|R|$, where $R = \{p(x, y) \in D \mid x_{low} \leq p.x \leq x_{high} \text{ AND } y_{low} \leq p.y \leq y_{high}\}$).

(1) [*3 marks*] Write a program to perform window queries based on the index you created in Task 1.
(2) [*3 marks*] Write a program to perform window queries based on the index you created in Task 2.
(3) [*5 mark*] Use the same 20 randomly generated window queries of different sizes and at different locations to search using the programs you developed in the above two subtasks (i.e., based on different indexing methods), to report (i) the number of points inside each query window, (ii) the number of points searched for each query respectively.

Task 4 [*4 marks*] You are required to write a report with no more than 6 pages (using this document as the template). Your goal in writing this report is to help the reader understand your design, your code, and your findings. The indexing structures and search algorithms much be clearly documented in plain language (if you prefer to use pseudocode, please make sure it is readable with proper comments). Note that the marks for this task will be allocated based on your report structure, clarity, readability, insightfulness and conciseness, while the assessment of the content in the report concerning each task above will be combined with the assessment of the corresponding tasks.
(1) [*2 marks*] To document any designs, explanations, or any notes for the previous tasks.
(2) [*2 mark*] To include the outputs and discussions for Task 1-3.

Notes:

1. In this assignment, you can use any programming language of your choice. No programming support will be provided in this course. No DBMS is needed. You will load the entire dataset into memory and perform all operations required in memory.
2. The query results (i.e., the number of points inside a query window) should be identical for the same query using different indexing structures, but the number of points compared can be different. This fact can be used to verify the correctness of your code.
3. You may be required to demonstrate and explain your programs in front of the TA. If there is such a need, you will be contacted by the TA to arrange a time and a way that is convenient for both you and the TA.
4. You are required to do this assignment independently, including developing all the code and doing the experiments. You should not copy the code from the Internet, any other sources, or from your classmates.

Submission guideline:

1. Late submission: unless approved by the lecturer or the TA in writing, every delay from one minute to 12-hours will incur a 25% deduction of your total marks for this assignment. That is, a delay of 2 days will lead to 0 marks for this assignment.

2. Submitted materials: should be compressed as a .zip file with student name as the file name

- Project report (up to 6 pages) in PDF format.
- Source code and a Readme file. Please document how we can run your code as well as how to install necessary packages, if any, in the Readme file. There is no need to include the dataset in your submission.
- Make sure your report and code contain your name and student ID.

3. Submission channel: on Canvas.

Warning: *This is an individual assignment. Collusion can be easily detected by software tools. Plagiarism will not be tolerated at HKUST. Please refer to Student Conduct and Academic Integrity regulations. If you are unclear about what level of discussions and help you can get for this assignment, please talk to the lecturer or the TA.*