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 2 [*Total: 30 marks*]

Due date: 11:59pm Saturday 30 April 2022
HKUST Canvas online submission only.

We will continue to use the POI dataset *D* used in Assignment 1.

Task 1 [*12 marks*] R-tree implementation.

(1) [*4 marks*] Write a program to create an R-tree index for point data in-memory. The fan-out of the tree should be *d* (i.e., a non-leaf node can have a maximum of *d* MBRs/subtrees), and each leaf node can contain a maximum of *n* points (i.e., the bucket size is *n*), where both *d* and *n* are user-given parameters. You can implement your program by looking at or using any code online (please make sure that the code is correct and suitable for this assignment, and you do understand the code! The source of the code, if you use the code from other sources, must be acknowledged in your report).

(2) [*4 marks*] Provide a concise outline of the algorithm you implement, with sufficient plain English comments such that your code can be easily understood by other people.

(3) [*4 marks*] Using the program developed above to create an R-tree index for the POI dataset *D*. You create a R-tree for the first half of *D*, and then for the entire *D*, and report for each case the following statistics with n=64 and 256 and d = 8 and 32 respectively (i.e., a total of 4 cases for the first half of *D* and then for the entire *D*):

   a. [*1 mark*] the height of your R-tree index.
   b. [*1 mark*] the numbers of non-leaf and leaf nodes.

Task 2 [*14 marks*] *k*NN search. For a query point *q*, find point *p* in *D* that is the nearest neighbor of *q*. L_2 distance is used in this task.

(1) [*3 marks*] Write a program that can find the nearest neighbor of a given query point by exhaustive search (i.e., checking the Euclidian distance from the query point to all the points in *D*).

(2) [*7 marks*] Write a program that can find the nearest neighbor of a given query point based on the R-tree implemented in Task 1 using the algorithm with the three pruning rules as discussed in this course.

(3) [*4 marks*] Generate 30 random query points, and run the two NN programs you implemented above to report the following statistics (please use n=256, d = 8 for your R-tree):

   a. [*1 mark*] The nearest neighbor result *p* and its L_2 distance to *q*.
   b. [*1 mark*] The running time to execute each query for the two algorithms (you should run your algorithms multiple times for each query and report the min/max/avg time).
   c. [*1 mark*] The number of points where the actual distance to *q* is calculated.

d. [*1 mark*] The number of MBRs that have been pruned out using each of the three pruning rules.

Task 3 [*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 must 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 Tasks 1-2.

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 including the R-tree.
2. 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.
3. You are required to do this assignment independently, including developing all the code and doing the experiments. You are allowed to copy the code from the Internet only for R-tree. Any other code should not be copied from the Internet, any other sources, or 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.*