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

# MSBD6000J Spatial and Multimedia Databases

*2021-2022 Fall*

**Version 1.0**

## Assignment 1 [*Total: 30 marks*]

**Due date:** 5pm Tuesday 26 October 2021
*HKUST Canvas online submission only.*

You are given a simplified real-world database *D* of points-of-interests (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.

In this assignment, 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. A search is done by first locating all necessary $r$ rectangles according to the query, and then checking the points in each relevant rectangle. The capacity of each bucket is fixed at 256 (that is, a bucket can hold no more than 256 points).

We will only process window queries. That is, for a given query rectangle represented as $Q = \{(x_{low}, y_{low}), (x_{high}, y_{high})\}$, find $R = \{p(x, y) \in D \mid x_{low} \leq p.x \leq x_{high}$ AND $y_{low} \leq p.y \leq y_{high}\}$.

This assignment only considers a simplified scenario where all the data are loaded into memory. That is, no disk-based operations will be considered.

**Task 1** [*1 mark*] Write a program to compute and report the MBR for the POI dataset *D*.

**Task 2** [*10 marks*] Create an in-memory index for D. The index cells are obtained following EXCELL idea. The number of cells must be $n$ x $n$, where $n$ is the minimum integer that ensures no bucket exceeds its capacity. That is, we have the equal-sized internals for the $x$-axis and $y$-axis.

(1) [*3 marks*] Write a program to create an EXCELL index. Report the value of $n$, and the numbers of the cells which contain 0, 1-25, 26-239, 240-255 and 256 points respectively.
(2) [5 *marks*] Write a program to perform window queries based on the index you created in the previous step (to return the number of points inside a query window).
(3) [2 *mark*] Generate 10 random window queries, run your search program, and report the number of points inside each query window, the number of index cells and the number of points searched for each query respectively.

**Task 3** [*10 marks*] Create an in-memory index for *D* as in Task 2, but this time the index cells are obtained following the quad-tree decompaction. Let $m$ be the minimum level of decompaction required such that no bucket exceeds its capacity.

(1) [*5 marks*] Write a program to create an index based on quad-tree decompaction. Report the value of *m*, and as in Task 2, the numbers of the cells which contain 0, 1-25, 26-239, 240-255 and 256 points respectively.

(2) [3 *marks*] As in Task 2, write a program to perform window queries based on the index you created in the previous step (to return the number of points inside a query window).

(3) [2 *mark*] Use the same 10 window queries used in Task 2, run your search program, and report the number of points inside each query window, the number of index cells and the number of points searched for each query respectively.

**Task 4** [*9 marks*] You are required to write to report with no more than 6 pages (using this document as the template).

(1) [*3 marks*] To document any designs, explanations, or notes for the previous tasks. The algorithm description for Task 3 is compulsory, and others are optional. Your goal is to help readers to understand your design and your code.

(2) [*2 marks*] To include the outputs to be reported in Task 1-3 in this report.

(3) [*4 marks*] To compare and contrast the two indexing approaches, focusing on the number of index cells, bucket space utilisation, search costs, and index maintenance overhead (i.e., when points are inserted or deleted).

**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 search algorithm in Task 3 can be identical to the search algorithm developed in Task 2 if you design index structures carefully. Therefore, it is possible to simply reuse your code. But this is up to you.
3. The query results (i.e., the number of points inside a query window) should be identical for the same query in Task 2 and Task 3, but the number of index cells and points visited can be different. This fact can be used to verify the correctness of your code.
4. Your report in Task 4 should be well structured and carefully written, and your marks for this report will be given based on correctness, completeness, insightfulness and conciseness.
5. Future submission about submitting your code will be provided by the teaching assistant (TA) later.
6. 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 which are convenient for both you and the TA.
7. You are required to do this assignment independently, including developing all the code. 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 20% deduction of your total marks for this assignment. That is, a delay of 2.5 days will lead to 0 marks for this assignment.

2. Submitted materials: should be compressed as a .zip file with student id 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 helps you can get for this assignment, please talk to the lecturer or the TA.*