RMIT UNIVERSITY

School of Science/ Computer Science and IT

COSC2636 Big Data Management

Assignment 1

Semester 1, 2019

---

1 Introduction

This is an individual assignment, to be submitted electronically using the Blackboard facility. A submission link will be enabled on blackboard closer to the submission date. It is due 23:59 Friday of Week 5, and contributes 15% towards the aggregate of 100 marks.

The objective of this assignment is to reinforce what you have learned in the lectures and tute/ lab sessions. Specifically, Assignment 1 is centered around the IEEE TKDE paper "INSPIRE: A Framework for Incremental Spatial Prefix Query Relaxation" to build the two-level indexes.

1.1 Plagiarism

All code or other material that is not original must be fully credited. That is, any material that is copied or derived from another source must be clearly identified as such and the original author must be identified. Sometimes students assi gether too closely, so that the students' separate solutio s were developed independently, they are r

Plagiarism is a very serious offence. Any submissions dete lagiarism will be deemed as an academic misconduct and harsh penalties apply. It is a s to allow their work to be plagiarised by another student. You should familiarize y website for Academic Integrity Policy, Procedures and Guidelines. (https://www.r nt-essentials/rights-and-responsibilities/academic-integrity) All work is to be done individually and plagiarism of any form will be dealt with according to the RMIT plagiarism policy.

1.2 What to Submit, When, and How

1.2.1 When

This assignment is due at 23:59 Friday of Week 5.

1.2.2 What

You should submit your implemented project as a zip file.
Naming convention: StudentNumber_A1.zip

The zip file should contain
- a folder, storing all your codes and a readme.txt describing how to run your code
- a report in pdf format, naming convention "report_studentnumber.pdf".
At the beginning of the report, write down your student number and name.

A report should include your understanding about each function used in the function "indexBuilding", such as achievement/functionality of each function, relationships among these functions and collaboration among

these functions to get the expected output. There is no page limit for the report but you should make your report as detailed as possible.

1.2.3 How

You are required to submit your solution electronically using the Canvas facility. A submission link will be enabled on Canvas closer to the submission date.

1.2.4 Penalties for late submissions

Late submissions of assignments will be penalised as follows. For 1 to 5 days late, a penalty of 10% (i.e. 10% out of total marks, not 10% out of your marks) per day. For assignments more than 5 days late, 100% penalty applies.

1.2.5 Special Consideration

If unexpected circumstances affect your ability to complete the assignment you can apply for special consideration. If you seek a short extension, you can directly contact the lecturer. For longer extensions, you must follow instructions provided at http://www1.rmit.edu.au/students/specialconsideration

1.3 Preparation Tasks

The code skeleton is provided in Canvas. Use the skeleton to implement functions missing in the class "buildIndex.BuildAll.java"

2 Task: Building the two-level inverted index (15 Marks)

You need to implement the function indexBuilding in BuildAll.java to build the two-level inverted index. The function locates at line 257-275.

The exemplar command is: sg.txt index 2 3 9 24 5 10

To check whether you built the index successfully, you should try to output information to see whether the statistics of your built index can match the correct one as follows:

building object database and Quadtree...
object database and quad tree build time : 3 second
number of sparse nodes in Quadtree: 3013
number of leaf nodes in Quadtree: 5320
number of level in Quadtree: 11
positional q-gram infrequent ratio : 2930 / 7323
q-gram token infrequent ratio : 8234 / 15078
infrequent inverted index building time : 0 seconds

building node-level and object-level inverted index...
node-level and object-level inverted index build time : 12 seconds
number of sparse nodes in Quadtree: 3013
number of leaf nodes in Quadtree: 5320
number of level in Quadtree: 11
infrequent positional q-gram inverted index size: 2930
first-Level positional q
second-Level position
infrequent q-gram toke
first-Level q-gram token inverted index size: 16725
second-Level q-gram token inverted index size: 44963

If your implementation is right, you can find the gener lder.

3 Grading criteria

We will use your program to generate indexes for other datasets. We will grade your program based on the correctness and the efficiency of your program, and also the report which contains your understanding about the program.