Term Project: Write-up Instructions

CS 386 Database Management

Prof. Daniel Miranker

Due: No later than 5:00 PM, May 2. Turn in Miranker's mailbox main office, Tay 2.144. Projects turned in late, but by 9:00 am May 5, will lose half a letter grade; (including all those slid under my door after 5:00 May 2. Later than that, one full letter grade.

1.0 Overview

At this juncture you should be near completion of the implementation component of your term project. The final report should have four primary sections.

I) A general discussion of metric-space indexing. You should relate this to the course material on multi-dimensional indexing and B-trees.

II) A specific discussion of the definition and implementation of your project.

III) Empirical results. (e.g. size, shape, balance of the tree with respect to loading the data set, look-up time, I/O, number of distance comparisons made)

IV) A tutorial/user’s guide to your project

2.0 Report Contents

The following is a recommended outline. Each section must start on a new page. Starred, “*”, sections are mandatory and their contents should be presented precisely according to the instructions.

*0. Cover page - title, your name, your partner’s name, if any, a URL to a page containing links to hypertext portions of your project and write-up a 100 to 300 word abstract.

1. Problem Definition

   • Introduction
     Explain the problem in general terms.
     Background
• Your own explanation of the problem addressed by your project
• Discuss your approach/solution

II. Engineering Aspects

A) Speak to the architecture of your project. (Figures containing UML sequence diagrams are an acceptable approach. If you don’t know what that means; don’t worry about it.)

B) Speak to the implementation aspects of each major component. If the major component comprises code you did not write (e.g. the m-tree package) speak to the ease of difficulty of using the code and make a qualitative assessment of that code.

*C) Tabulate (make a table) engineering properties of the different components. e.g. for each component list, number of files, number of lines of code, number of classes...

*D) Document an execution of your program.

III Empirical Results

A) Present an overview of the parameterizations you chose to evaluate. Try to develop specific hypothesis about the data structure, e.g. “larger fan-outs at the nodes better accommodates high-dimensionality”. Relate the parameterization to an assessment of the hypothesis. For inspiration you may want to look at the feedback provided by amdb package associate with GiST (see http://gist.cs.berkeley.edu) or review related technical papers.

Explain the larger structure of your test environment and test work load.

B) Present quantitative results.

C) Discuss

*IV Tutorial

The primary purpose of the tutorial section of your report is to provide a set of instructions on how to step through your code and see it work. There will be no demo of your code. I will follow the steps of your tutorial until completed or the code crashes, which ever comes first. Implementation points will be awarded on that basis.

V Discussion and Conclusion

This section is your best opportunity to express critical thought on the project. It is dubious whether anything less than half a full page will fulfill this section.