# CPSC 404: Advanced Relational Databases
# Quiz 1

Laks V.S. Lakshmanan

**Date**: Tuesday, February 24, 2009. **3:35-4:25 pm**.

# IMPORTANT INSTRUCTIONS:

- Everyone must copy the following honor code verbiage in their answer booklet and sign their name:
  "I am aware of what constitutes academic misconduct and the disciplinary actions that may be taken against it, and agree not to cheat."
- Exams without a signed honor code will not be marked and the student will be presumed to have missed the quiz effectively.
- Be sure to print **your name** and **your student ID** clearly and prominently on your answer booklet.
- There is an optional **bonus question** at the end, and is worth 10 points. This is on top of the 100 points that the other questions are worth.
- Answer **all** questions, except the bonus question is optional. I suggest you attempt it after answering other questions.
- **Show all your work**, including all the steps. **State your assumptions**, if any, clearly. Answers without proper explanation or details will get a low credit.
- At the end of the quiz, you must hand in *both* your answer booklet and this quiz (i.e., the questions) and remain seated until all exams have been collected.
- Beside each question, you will find not only the maximum marks for the question, but also a suggested time. This budgeting assumes 45 minutes for writing the exam and 5 minutes for reviewing your answers.
- Questions continue overleaf.

**Question 1[50%](suggested time: 23 min.)**
**(B+trees: (a)-(d); Extendible hashing: (e).)**
Consider a file of 2 million records, 1000 bytes each. The size of a primary key value is 32 bytes and that of a disk pointer as well as that of a record id (rid) is 8 bytes. Each page is 8K. Each data entry node as well as each index node is implemented as a page.

(a) [**8%**] Suppose the file is **not** sorted on its primary key and you create a B+tree index on the primary key, with data entries of the form (key, rid) (i.e., the so-called alternative 2). Will this be a clustered or unclustered index?
(b) [**8%**] How many data entries are there in your index?
(c) [**10%**] How many leaf pages are there? (Recall the contents of the leaf pages are data entries as

described in (a). Assume pages are filled to capacity.)
(d) [**12%**] Assuming each node of the B+tree is about 70% full, what would be the height of this B+tree?
(e) [**12%**] Suppose we start with an extendible hash structure with 2 empty buckets, corresponding to 0 and 1. Each bucket can hold up to 2 data entries. The hash function is $h(k) = k \ mod \ 8$. Suppose the following data entries are inserted in succession:
16*, 33*, 50*, 67*; 36*; 165*; 326*, 135*; 24*, 31*. Show the hash structure (including directory and buckets, along with local and global depth) after **every underlined sequence of insertions**.

**Question 2**[**50%**] **(suggested time: 22 min.)**
(**Sorting.**)
Consider a file with 22,000 disk pages. Suppose a buffer with 22 pages is available and you apply the external sorting algorithm discussed in class to sort this file, *incorporating cylindrification, prefetching*, and *double buffering*. Assume that in each iteration in phase II, we aggressively *merge as many sorted sublists as possible*, while using double buffering. Now, answer the following questions.

(a) [**10%**] How many sorted sublists (SSLs) will be produced in phase I?
(b) [**10%**] What is the maximum number of SSLs you can merge in one iteration in phase II, while employing double buffering?
(c) [**10%**] How many iterations (passes) are needed in phase II to complete the merging?
(d) [**10%**] What is the total number of page reads/writes over both phases?
(e) [**10%**] xWhat is the total number of random accesses incurred in phase II?

**Bonus Extrea Credit Question** [**10%**] (**suggested time: if you finish other questions quicker, 5+ min.**)
For Question 2 above, assume each random access costs 25 ms and each page costs 0.5 ms to read or write. Then what is the optimal number of SSLs that you'd merge in one iteration in Phase II, to minimize overall sorting cost? You need not prove optimality formally, but need to show your work and reasoning involved in arriving at your conclusion.