CPSC 404: Advanced Relational Databases
Quiz 1 – Version B
Laks V.S. Lakshmanan

Date: Monday, October 7, 2002. **Duration: 45 minutes**.

IMPORTANT INSTRUCTIONS:

- Everyone must copy the following honor code verbiage in their answer booklet and sign: "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 write your name in block letters and print your student ID.
- Answer **all** 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).

Question 1. Consider a disk with the following parameters:

(i) disk rotation: 2,000 rpm. (ii) arm movement takes 1.002*n ms, to move over n cylinders. (iii) size of each sector 1,024 bytes. (iv) number of platters = number of surfaces = 10; (v) number of sectors per track 128; (vi) number of cylinders 2^{12}.

(a) What is the total capacity of the disk in bytes?

(b) What are the minimum, maximum, and average seek times?

(c) What are the maximum, and average rotational latencies?

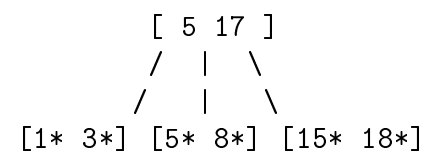
Question 2. Consider a file of 2 million records, 400 bytes each. Assume a block size of 4K and the parameters of the disk described in Question 1. Suppose we wish to sort this file on its primary key, using a buffer of size 20M. Suppose in the second phase, all sorted sublists are merged in one shot. Then estimate the total time taken for sorting this file, clearly showing how much time is spent on reading/writing the file versus how much on making random I/Os. Your calculations must take into account cylindrification, prefetching, and double buffering.

Question 3. Consider the above file of 2 million records, 400 bytes each. The size of a primary key value is 16 bytes and that of a pointer is 4 bytes. We use the records themselves as data entries.

(a) Estimate the maximum and minimum height of a B^+-tree for this file.

(b) Estimate the minimum number of blocks needed for holding this B^+-tree index.

(c) Consider the B+ tree shown below (order d = 1).



Show the index after inserting 7* and also after inserting 9*; then show it after deleting 1*; and then again after inserting 3*.

(d) Consider the following linear hashing index.

h_1	h_0	primary pages	overflow pages
000	00	[64*|8*|48*|]	
001	01	[17*|33*|81*|41*]	
010	10	[22*|34*|26*|70*]	
011	11	[43*|47*|23*|27*]	
100	00	[52*|44*| |]	

The Next pointer currently points to bucket number 1. The current round number (i.e., Level) is 0. Answer the following questions. Assume that every time there is an overflow, the current bucket is split.

(i) Identify those buckets that have yet to be split in the current round.

(ii) Give an example of an insertion that would cause the current bucket to be split. What happens to the Next pointer?

(iii) After the insertion in question (ii), suppose the data entry 42* is inserted. Show what happens to the linear hash structure, including the Next pointer.