The assignment is to be turned in before Midnight (by 11:59pm) on January 21st, 2016. You should turn in the solutions to this assignment as a pdf file through the TEACH website. The solutions should be produced using editing software programs, such as LaTeX or Word, otherwise they will not be graded.

1: Operating System vs. RDBMS (1 point)

Explain the term prefetching and describe one of its benefits. Why should database management system do its own prefetching rather than relying on the prefetching provided by operating system.

Solution: Because most block references in a DBMS environment are with a known reference pattern, the buffer manager can anticipate the next several block requests and fetch the corresponding pages into memory before the pages are requested. This is prefetching. By prefetching, the blocks are available in the buffer pool when they are requested.

2: Operating System vs. RDBMS (0.5 point)

Database management systems may support multiple (concurrent) users by assigning a separate operating system process to each user. Briefly explain the potential problems of this approach.

Solution: Because database processes have a great deal of state information, it may take a relatively long time for the OS process scheduler to switch these processes. Further, if the descheduled process has locks on some data items in the database, other database processes have to wait until the OS scheduler picks the descheduled process for execution.

3: B+ tree Indexing (3 point)

Using B+ trees with degrees of two (2) whose keys are integer values, answer the following questions. A B+ tree with a single node has height of 1.

1. Give an example of a B+ tree whose height changes from 2 to 3 when the value 25 is inserted. Show your structure before and after the insertion.

2. Give an example of a B+ tree in which the deletion of the value 25 causes a merge of two nodes but without altering the height of the tree.

Solution:

1. One solution is a B+ tree has a root and five leaf nodes. The root node contains four keys of [10, 20, 30, 40]. The five leaf nodes, from left to right, contain [2,6, [10,13,16,17], [20,21,23,28], [31,32,36,38], and [43,54,69,87]. After inserting 25, the resulting B+ tree will have one root node, two internal nodes, and six leaf nodes. The root node contains a single key 23. The internal nodes, from left to right, contain [10,20] and [30,40]. The leaf nodes, from left to right, contain keys [2,6, [10,13,16,17], [20,21], [23,25,28], [31,32,36,38], and [43,54,69,87].
2. One solution is a B+ tree has a root and three leaf nodes. The root node contains four keys of [10, 20]. The three leaf nodes, from left to right, contain [2,6] , [10,13,16], and [20,25]. After deleting 25, the resulting B+ tree will have one root node and two leaf nodes. The root node contains a single key of [10]. The two leaf nodes, from left to right, contain [2,6] and [10,13,16,20].

4: B+ tree Indexing (0.5 point)

To reduce the number of I/O access in index search, each B+ tree node should fit in a block. Let the key value and record pointer for a B+ tree be 32 and 64 bytes, respectively. If the block size is 16384 bytes, what should be the minimum degree of the B+ tree?

Solution: \(2d \times 32 + (2d + 1) \times 64 \leq 16384\).