COMP 530 Quiz #2

My name is _____

Circle the best answer for each question.

(1) Swizzling refers to:

(a) The process of converting “database” pointers to “OS” or “regular” pointers
(b) The process of converting “OS” pointers to “database” pointers
(c) The process of dereferencing an OS pointer in a database
(d) The process of dereferencing a database pointer in a database

(2) If the TPMMS algorithm has R pages of memory to operate with, about how big is the largest file that the algorithm can sort?

(a) R^(1/2) pages
(b) R pages
(c) R^(3/2) pages
(d) R^2 pages

(3) In class, we mentioned several ways for maintaining a sorted file. One is to use an overflow file, typically buffered in RAM, that is periodically merged with the sorted file. Another method that we didn’t discuss to build some empty space into the sorted file (“holes” in the pages) in order to absorb new insertions. Which of the following statements about the two options is incorrect?

(a) The overflow file option requires time O(n^2) to insert n records into the sorted file.
(b) Both options will require a complete scan/rebuild of the sorted file periodically.
(c) It is not possible to run a binary search when there are holes in pages.
(d) The overflow file option is more space-efficient.

(4) Why might the the “holes” option from question 5 more vulnerable to hotspots when processing insertions?

(a) It requires a binary search for every insertion when there is a hotspot
(b) If the insertion hotspot fills up a specific portion of the file, then I end up inserting the new record and sliding data over until I find some empty space
(c) It is not possible to keep the file sorted if there is a hotspot
(d) The “holes” option will leave more of the space in the file empty if there is a hotspot

(5) For what sort of database would a heap be the poorest choice?

(a) One characterized by a constant stream of queries returning most of the database’s data
(b) One characterized by a constant stream of inserts
(c) One characterized by a constant stream of range queries accepting 1% of the file’s records

(6) For what type of keys would a B+-Tree have the shortest path length from root to leaf?

(a) Long strings
(b) Double precision floating point numbers
(c) Single precision floating point numbers
(d) This question is silly; the path length does not depend on the key type

(7) What is the primary reason for the links across the leaves of a B+-Tree?

(a) To facilitate the indexing of keys that can be repeated in the table
(b) To make it possible to answer range queries, since without such links, they cannot be evaluated
(c) To make range queries really fast
(d) To allow concurrent operations to perform range queries without holding locks high in the tree

(8) A B+-Tree can have differing path lengths from root to leaf

(a) True
(b) False

(9) What is the minimum fill factor in any B+-Tree node other than the root?

(a) 33%
(b) 50%
(c) 75%
(d) 100%

(10) Why do we prefer B+-Trees to classical binary trees for block-oriented storage?

(a) Because B+-Trees have smaller nodes than classical binary trees
(b) Because classical binary trees have no Big-O guarantees once you have to be able to process insertions
(c) Because B+-Trees are designed to use most of the space on a page, meaning we don't waste bytes that are transferred into RAM
(d) Because B+-Trees send to be smaller than classical binary trees