CS4432: Database Systems II - Homework 2

April 3, 2025

Total Points: 100

Note: Please do it individually. In order to get a question graded, you will need to show your work.

Release Date: 04/03/2025 (12:30 PM)
Due Date: 04/10/2025 (11:59 PM)

Submission Instruction

1. You may use any tool you like to write your answers, especially for drawing the B-tree.
2. Convert your file(s) to .pdf and upload it.
3. Include your name inside the file.
4. Submit your work on Canvas.
5. Late Submission Policy: Follows the policy posted on the course website.

Generative AI Usage Instruction

You are permitted to use generative artificial intelligence tools (such as ChatGPT, etc.) to assist in completing the assignment. However, when using these tools, you must clearly state the following in your assignment:

1. How the generative artificial intelligence tool was used.
2. How the generated answers were verified to be correct.

1 Problem 1: External Sorting (15 Points)

Assume a file consisting of 200 blocks, which we need to sort.

(a) Assume we only have 5 memory buffers available

- What is the total number of I/Os to sort the entire file?
- How many sorted runs (lists) are there after the 1st merge step?
- How many sorted runs (lists) are there after the 2nd merge step?

(b) Assume we only have 8 memory buffers available

- What is the total number of I/Os to sort the entire file?
- How many sorted runs (lists) are there after the 1st merge step?
- How many sorted runs (lists) are there after the 2nd merge step?

2 Problem 2: Indexing (30 Points)

Consider a data file R consisting of 1,000,000 blocks that are contiguous on disk. Each block contains 20 records, i.e., the total number of records in R is (20 * 10^6). Each record consists of attributes (K1, K2, ...). Let K1 be the primary key of the relation, and hence R's records are physically sorted by K1. Also, let K2 be another attribute of R (unsorted attribute). Assume the size of K1 and K2 attributes is 20 bytes each, a record pointer is 8 bytes long, and a block is 8KB (8 * 1024 bytes). Assume no single record (data record or index record) can be divided across two blocks.

1. Is it possible to construct a dense sequential index (1-level) on K1 over R? Describe the layout, and how large (how many blocks) will the index be?
2. Is it possible to construct a sparse sequential index (1-level) on K1 over R? Describe the layout, and how large (how many blocks) will the index be?
3. Is it possible to construct a dense sequential index (1-level) on K2 over R? Describe the layout, and how large (how many blocks) will the index be?
4. Is it possible to construct a sparse sequential index (1-level) on K2 over R? Describe the layout, and how large (how many blocks) will the index be?
5. Is it possible to build a second-level index on the one built in 1? If yes, what will be the size of the index (how many blocks)? Report the size of the second-level alone and the total index size (both levels).

6. Is it possible to build a second-level index on the one built in 2? If yes, what will be the size of the index (how many blocks)? Report the size of the second-level alone and the total index size (both levels).

3 Problem 3: B+ Tree Indexing (30 Points)

Consider the B+ tree in Figure 14.13 from the course textbook (Page 636). Describe how each of the following operations would proceed. If it modifies the tree, draw the revised tree. Assume that an insert that does not find space in its node will trigger a splitting of this node. You can make additional assumptions, but you must always spell them out.

Assume the minimum number of keys allowed in an internal node is Floor(node size / 2) = 1

1. Lookup record with search key 35. Indicate which index pages are accessed? Write them in the order of their access. E.g., label each index node with a label like "N1", "N2", ..., and the specify the sequence of touched nodes to search for key 35.

2. Lookup all records within the range [9, 21]. Indicate which index pages are accessed? As in 1, write them in the order of their access.

3. Insert a record with key 4. Show the modified tree.

4. To the tree after the above insertion, insert record with key 14, then record with key 15, then record with key 16. Show the tree after every insert.

5. After the insertions in steps 3 & 4, lookup records within the range [6, 13]. Indicate which index pages are accessed? Write them in the order of their access.

6. After the insertions in 3 & 4, delete the record with key 23. Show the modified tree.