Homework #2 (by Arvin Wu)
Due: Friday February 17, 2023 @ 11:59pm

IMPORTANT:
- Enter all of your answers into Gradescope by 11:59pm on Friday February 17, 2023.
- Plagiarism: Homework may be discussed with other students, but all homework is to be completed individually.

For your information:
- Graded out of 100 points; 4 questions total
- Rough time estimate: ≈4-6 hours (1-1.5 hours for each question)

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage Models</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Cuckoo Hashing</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>Extendible Hashing</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>B+Tree</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
Question 1: Storage Models ...................................... [16 points]

Consider a database with a single table $T$($course_id$, $course_name$, instructor, class_size, hrs_per_week), where course_id is the primary key, and all attributes are the same fixed width. Suppose $T$ has 5,000 tuples that fit into 500 pages. Ignore any additional storage overhead for the table (e.g., page headers, tuple headers). Additionally, you should make the following assumptions:

- The DBMS does not have any additional meta-data (e.g., sort order, zone maps).
- $T$ does not have any indexes (including for primary key course_id)
- None of $T$’s pages are already in the buffer pool.
- Content-wise, the tuples of $T$ will make each query run the longest possible (this assumption is critical for solving part (a))
- The tuples of $T$ can be in any order (this assumption is critical for solving part (b) when you compute the minimum versus maximum number of pages that the DBMS will potentially have to read)

(a) Consider the following query:

$$\text{SELECT MAX(hrs\_per\_week) FROM T WHERE class\_size > 10;}$$

i. [3 points] Suppose the DBMS uses the decomposition storage model (DSM) with implicit offsets. How many pages will the DBMS potentially have to read from disk to answer this query? (Keep in mind our assumption about the contents of $T$!)

- $\square$ 1-100  $\square$ 101-200  $\square$ 201-300  $\square$ 301-500  $\square$ $\geq$ 501  $\square$ Not possible to determine

ii. [3 points] Suppose the DBMS uses the N-ary storage model (NSM). How many pages will the DBMS potentially have to read from disk to answer this query? (Keep in mind our assumption about the contents of $T$!)

- $\square$ 1-100  $\square$ 101-200  $\square$ 201-300  $\square$ 301-500  $\square$ $\geq$ 501  $\square$ Not possible to determine

(b) Now consider the following query:

$$\text{SELECT course\_name, instructor, class\_size FROM T WHERE course\_id = 15445 OR course\_id = 15645;}$$

i. Suppose the DBMS uses the decomposition storage model (DSM) with implicit offsets.

a) [3 points] What is the minimum number of pages that the DBMS will potentially have to read from disk to answer this query?

- $\square$ 1  $\square$ 2-4  $\square$ 5-100  $\square$ 101-200  $\square$ 201-500  $\square$ $\geq$ 501

Question 1 continues...
\( \beta \) [3 points] What is the maximum number of pages that the DBMS will potentially have to read from disk to answer this query?
- [ ] 1
- [ ] 2-4
- [ ] 5-100
- [ ] 101-200
- [ ] 201-500
- [ ] \( \geq 501 \)
- [ ] Not possible to determine

ii. Suppose the DBMS uses the N-ary storage model (NSM).
\( \alpha \) [2 points] What is the minimum number of pages that the DBMS will potentially have to read from disk to answer this query?
- [ ] 1
- [ ] 2-4
- [ ] 5-100
- [ ] 101-200
- [ ] 201-500
- [ ] \( \geq 501 \)
- [ ] Not possible to determine

\( \beta \) [2 points] What is the maximum number of pages that the DBMS will potentially have to read from disk to answer this query?
- [ ] 1
- [ ] 2-4
- [ ] 5-100
- [ ] 101-200
- [ ] 201-500
- [ ] \( \geq 501 \)
- [ ] Not possible to determine
Question 2: Cuckoo Hashing

Consider the following cuckoo hashing schema:

1. Both tables have a size of 4.
2. The hashing function of the first table returns the fourth and third least significant bits:
   \[ h_1(x) = (x >> 2) \& \text{0b11}. \]
3. The hashing function of the second table returns the least significant two bits:
   \[ h_2(x) = x \& \text{0b11}. \]
4. When inserting, try table 1 first.
5. When replacement is necessary, first select an element in the second table.
6. The original entries in the table are shown in the figure below.

![Figure 1: Initial contents of the hash tables.](image)

(a) [1 point] Select the sequence of insert operations that results in the initial state.
- Insert 16, insert 3
- Insert 3, insert 16
- None of the above

Question 2 continues...
(b) Insert key 8 and then delete 16. Select the value in each entry of the resulting two tables.

i. Table 1
- α) [1 point] Entry 0 (0b00) □ 3 □ 8 □ Empty
- β) [1 point] Entry 1 (0b01) □ 3 □ 8 □ Empty
- γ) [1 point] Entry 2 (0b10) □ 3 □ 8 □ Empty
- δ) [1 point] Entry 3 (0b11) □ 3 □ 8 □ Empty

ii. Table 2
- α) [1 point] Entry 0 (0b00) □ 3 □ 8 □ Empty
- β) [1 point] Entry 1 (0b01) □ 3 □ 8 □ Empty
- γ) [1 point] Entry 2 (0b10) □ 3 □ 8 □ Empty
- δ) [1 point] Entry 3 (0b11) □ 3 □ 8 □ Empty

(c) After the changes from part (b), insert key 27 and then insert 4. Select the value in each entry of the resulting two tables.

i. Table 1
- α) [1 point] Entry 0 (0b00) □ 3 □ 8 □ 27 □ 4 □ Empty
- β) [1 point] Entry 1 (0b01) □ 3 □ 8 □ 27 □ 4 □ Empty
- γ) [1 point] Entry 2 (0b10) □ 3 □ 8 □ 27 □ 4 □ Empty
- δ) [1 point] Entry 3 (0b11) □ 3 □ 8 □ 27 □ 4 □ Empty

ii. Table 2
- α) [1 point] Entry 0 (0b00) □ 3 □ 8 □ 27 □ 4 □ Empty
- β) [1 point] Entry 1 (0b01) □ 3 □ 8 □ 27 □ 4 □ Empty
- γ) [1 point] Entry 2 (0b10) □ 3 □ 8 □ 27 □ 4 □ Empty
- δ) [1 point] Entry 3 (0b11) □ 3 □ 8 □ 27 □ 4 □ Empty

Question 2 continues...
(d) After the changes from parts (b) and (c), insert key 19 and then delete 27. Select the value in each entry of the resulting two tables.

i. Table 1
   α) [1 point] Entry 0 (0b00) □ 3 □ 8 □ 4 □ 19 □ Empty
   β) [1 point] Entry 1 (0b01) □ 3 □ 8 □ 4 □ 19 □ Empty
   γ) [1 point] Entry 2 (0b10) □ 3 □ 8 □ 4 □ 19 □ Empty
   δ) [1 point] Entry 3 (0b11) □ 3 □ 8 □ 4 □ 19 □ Empty

ii. Table 2
   α) [1 point] Entry 0 (0b00) □ 3 □ 8 □ 4 □ 19 □ Empty
   β) [1 point] Entry 1 (0b01) □ 3 □ 8 □ 4 □ 19 □ Empty
   γ) [1 point] Entry 2 (0b10) □ 3 □ 8 □ 4 □ 19 □ Empty
   δ) [1 point] Entry 3 (0b11) □ 3 □ 8 □ 4 □ 19 □ Empty

(e) [2 points] What is the smallest key that potentially causes an infinite loop given the table that results from part (d)?
   □ 33 □ 34 □ 35 □ 36 □ 37 □ 51 □ None of the above
Question 3: Extendible Hashing ............................................. [22 points]

Consider an extendible hashing structure such that:

- Each bucket can hold up to two records.
- The hashing function uses the lowest \( g \) bits, where \( g \) is the global depth.
- A new extendible hashing structure is initialized with \( g = 0 \) and one empty bucket

(a) Starting from an empty table, insert keys 6, 1.

i. [1 point] What is the global depth of the resulting table?
   □ 0 □ 1 □ 2 □ 3 □ 4 □ None of the above

ii. [1 point] What is the local depth of the bucket containing 6?
   □ 0 □ 1 □ 2 □ 3 □ 4 □ None of the above

(b) Starting from the result in (a), you insert keys 17, 10.

i. [2 points] What is the global depth of the resulting table?
   □ 0 □ 1 □ 2 □ 3 □ 4 □ None of the above

ii. [1 point] What is the local depth of the bucket containing 10?
   □ 0 □ 1 □ 2 □ 3 □ 4 □ None of the above

(c) Starting from the result in (b), you insert key 25.

i. [3 points] What is the global depth of the resulting table?
   □ 0 □ 1 □ 2 □ 3 □ 4 □ None of the above

ii. [3 points] What is the local depth of the bucket containing 25?
   □ 0 □ 1 □ 2 □ 3 □ 4 □ None of the above

iii. [2 points] What is the local depth of the bucket containing 17?
   □ 0 □ 1 □ 2 □ 3 □ 4 □ None of the above

iv. [1 point] What is the local depth of the bucket containing 6?
   □ 0 □ 1 □ 2 □ 3 □ 4 □ None of the above

v. [1 point] Which value, if inserted, will hash to the same bucket as the bucket containing key 10?
   □ 4 □ 7 □ 9 □ 13 □ None of the above

(d) [2 points] Starting from the result in (c), which key(s), if inserted next, will cause a split that doubles the table’s size?
   □ 5 □ 12 □ 24 □ 29 □ 33 □ None of the above

(e) [2 points] Starting from the result in (c), which key(s), if inserted next, will cause a split without doubling the table’s size?
   □ 5 □ 12 □ 24 □ 29 □ 33 □ None of the above

(f) [3 points] Starting from an empty table, insert keys 32, 64, 128, 256. What is the global depth of the resulting table?
   □ 4 □ 5 □ 6 □ 7 □ 8 □ \( \geq 9 \)
Question 4: B+Tree ....................................................... [35 points]
Consider the following B+tree.

When answering the following questions, be sure to follow the procedures described in class and in your textbook. You can make the following assumptions:

- A left pointer in an internal node guides towards keys < than its corresponding key, while a right pointer guides towards keys ≥.
- A leaf node underflows when the number of keys goes below ⌈d−1/2⌉.
- An internal node underflows when the number of pointers goes below ⌈d/2⌉.

Figure 2: B+ Tree of order \( d = 4 \) and height \( h = 2 \).

Note that B+ tree diagrams for this problem omit leaf pointers for convenience. The leaves of actual B+ trees are linked together via pointers, forming a singly linked list allowing for quick traversal through all keys.

Question 4 continues...
(a) [5 points] Insert 23 into the B+tree. Select the resulting tree.

A)

B)

C)

D)

Question 4 continues…
(b) [5 points] Starting with the tree that results from (a), insert 15°. Select the resulting tree.

□ A)

![Image A]

□ B)

![Image B]

□ C)

![Image C]

□ D)

![Image D]

Question 4 continues . . .
(c) [5 points] Starting with the tree that results from (b), delete 40. Select the resulting tree.

- A)

- B)

- C)

- D)

Question 4 continues...
(d) [5 points] Starting with the tree that results from (c), insert $16^*$. Select the resulting tree.

- A)

- B)

- C)

- D)

Question 4 continues...
(e) [5 points] Finally, starting with the tree that results from (d), delete $5^\ast$. Select the resulting tree.

- A)

- B)

- C)

- D)

Question 4 continues...
(f) The B+Tree shown in Figure 3 is invalid. That is, its nodes violate the correctness properties of B+Trees that we discussed in class. If the tree is invalid, select all the properties that are violated for each node. If the node is valid, then select ‘None’. There will be no partial credit for missing violations.

Note:

• If a node’s subtrees are not the same height, the balance property is violated at that node only.
• If a node’s subtrees contain values not in the range specified by the node’s separator keys, the separator keys property is violated at that node.

1. [2 points] Which properties are violated by Leaf 1?
   - Key order property
   - Half-full property
   - Balance property
   - Separator keys
   - None

2. [2 points] Which properties are violated by Leaf 2?
   - Key order property
   - Half-full property
   - Balance property
   - Separator keys
   - None

3. [2 points] Which properties are violated by Internal Node 1?
   - Key order property
   - Half-full property
   - Balance property
   - Separator keys
   - None

4. [2 points] Which properties are violated by Internal Node 2?
   - Key order property
   - Half-full property
   - Balance property
   - Separator keys
   - None

5. [2 points] Which properties are violated by Root?
   - Key order property
   - Half-full property
   - Balance property
   - Separator keys
   - None