EECS 484: Sample Final Exam #2

Please ***print*** your name and uniqname below.

Name:
Uniqname:

Instructions:

1. You have 2 hours to complete this exam. The total number of points on this exam is **100**.
2. The questions vary widely in difficulty. Watch your time carefully to avoid getting stuck on a hard question at the expense of an easier question.
3. Some of the questions on this exam require you to provide an explanation. ***You will be graded on the clarity of your explanation, in addition to the correctness of your answer.***
4. At the end of the exam, sign the Honor Pledge given below.

Honor Code Pledge: I have neither given nor received assistance in this exam, nor have I concealed any violation of the Honor Code.

Signature:

(For Instructor's Use)

Q1: Storage and Files (16 points)	
Q2: Indexes (20 points)	
Q3: Query Evaluation (30 points)	
Q4: Transactions, Locking (14 points)	
Q5: Recovery (20 points)	
TOTAL	

Question 1. (16 points) Storage & Files

(a) (4 points) Consider a typical magnetic disk. Briefly explain the difference between a random I/O and a sequential I/O in terms of what the disk must do for each.

(b) (6 points) In the Minirel buffer manager, for each buffer frame, a variable `pinCount` is used to track the number of transactions that use the page. Suppose that you are re-designing the system. Rather than storing an integer `pinCount` for each frame, would it be appropriate to replace it with a boolean value `isPinned`? Discuss whether this is a good idea. *Explain your answer.*

(c) (6 points) Suppose that you have a relation S, and that your query workload requires you to scan S three times. Suppose that S has 7 pages, and that your buffer pool contains 4 frames. How many I/Os are required if you use each of the following page replacement algorithms? *In both cases, assume that the buffer pool is initially empty.*

Least Recently Used (LRU): _____

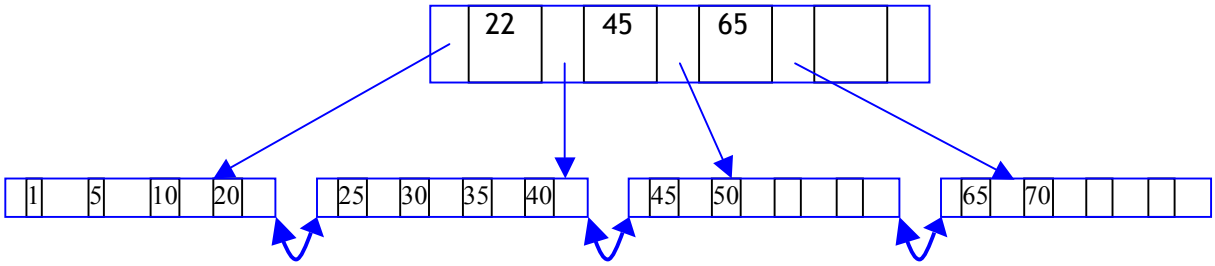
Most Recently Used (MRU): _____

Question 2. (20 points) Indexes

(a) (4 points) Consider the predicate "(a = 5 AND b = 20) OR (a = 5 AND c < 30)". Which of the following access methods can be used (by itself) to process a query containing this predicate? (Select all that apply.) *Hint: first transform the predicate to CNF.*

[ ] Hash index on a	[ ] B+Tree on a	[ ] B+Tree on c
[ ] B+Tree on <c, b>	[ ] File Scan	[ ] Hash index on b

(b) (2 points) Consider the B+-Tree shown below. Assume the convention that data entries on the left are strictly less than the key value, and data entries on the right are greater than or equal to the key value. Assume that the leaves store data entries as (key, rid) pairs, and you do NOT use leaf redistribution for inserts. Draw the tree after inserting the data entry with search key value **32**.

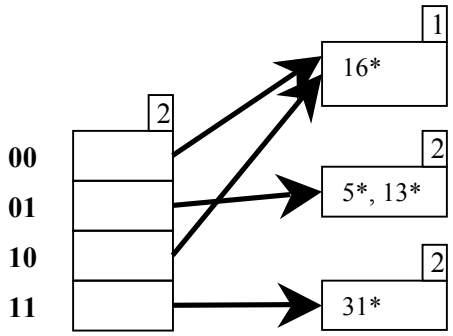


(c) (4 points) Show the tree after additionally inserting (in this order) data entries with key values **7** and **12**.

(d) (4 points) Consider a B+Tree that contains a total of 100,000 data entries in the leaves. Suppose that 50 data entries can fit on a page. Suppose that (on average), a non-leaf node in the B+Tree has 70 children. What is the height of the B+Tree? *Show your work for full credit.*

(e) (6 points) Consider the following instance of an extendible hash index. The directory is shown on the left, and the pages containing data entries are shown in the right. The global depth (2) is shown next to the directory, and a local depth is given for each page. Assume (for simplicity) that the hash function is h(K) = K, and that at most 2 data entries fit on a page.

Re-draw the index to show how it would appear after inserting (in this order) the data entries **4*** and **1*.**



Question 3. (30 points) Query Evaluation & Optimization

(a) (12 points) This question relates to external sorting. Suppose that you are implementing an external mergesort routine for a new open source DBMS, and you are trying to decide whether or not to use the *double-buffering* optimization. For this problem, you can make the following simplifying assumptions:

- The DBMS uses a page size of 1024 bytes
- The buffer pool can always hold exactly 10,000 pages
- You are always sorting files that are exactly 1 GB.

Do you expect double-buffering to be helpful in this scenario? Explain your answer and show your work. (Hint: *How many passes will be required with and without double-buffering?)*

(b) (5 points) Consider joining two relations, R and S, using a Block Nested Loops join, in which R is the *outer* relation. Suppose that R contains 1,000 pages, S contains 5,000 pages, and that you have 200 pages of memory available to use in processing the join. How many times do you have to scan S? Explain your answer and show your work.

(c) (3 points) Consider a relational schema with two tables: A(a1,a2,a3) and B(b1, b2, b3), and the following relational algebra expression. To which of the given expressions is it equivalent? (Select all that apply.)

$\pi_{a3}\ (\sigma_{b2 = 5}\ (A \bowtie_{a1 = b1} B))$

[ ] $\pi_{a3}\ (\sigma_{b2 = 5}\ (\sigma_{a1 = b1}\ (\sigma_{b2 = 5\ V\ b2 = 10}\ (A \times B))))$
[ ] $\pi_{a3}\ (\sigma_{b2 = 5}\ (\pi_{a3}\ (A \bowtie_{a1 = b1} B)))$
[ ] $\pi_{a3}\ ((\sigma_{b2 = 5}\ B) \bowtie_{a1 = b1} A))$

(d) (6 points) Recall that cardinality estimation is an important part of query optimization. Using statistics stored in the system catalogs, the query optimizer estimates the number of tuples that will pass a given selection condition. Describe *two* (2) potential shortcomings of System R's approach to cardinality estimation. (Feel free to illustrate using examples.)

(e) (4 points) Consider the following SQL query.

```
SELECT S.sname, P.pname
FROM Suppliers S, Parts P, Supply Y
WHERE S.sid = Y.sid AND Y.pid = P.pid AND S.city = 'Ann Arbor'
```

How many different join orders does the System R query optimizer consider (assuming cross-products are disallowed) when deciding how to process the query? List each of these join orders.

Question 4. (14 points) Transactions and Locking

Consider the following schedule, which involves three transactions (T1, T2, and T3) and three data objects (X, Y, and Z).

T1	**T2**	**T3**
R(X)		
W(X)		
		W(Z)
	R(X)	
	R(Y)	
	W(Y)	
R(Z)		
	Commit	
Commit		
		Abort

(a) (5 points) Is the schedule *serializable*? Briefly explain your answer.

(b) (3 points) Is the schedule *recoverable*? Briefly explain your answer.

(c) (2 points) The Strict 2-Phase Locking protocol (Strict 2PL) is guaranteed to provide a serializable schedule.
[ ] True [ ] False

(d) (4 points) Suppose that a transaction T1 currently holds an exclusive (X) lock on record R101. Briefly explain what the lock manager must do if another transaction T2 requests a shared (S) lock on record R101.

Question 5. (20 points) Recovery

Consider the following log and the Aries recovery system.

<table>
<tr><th>LSN</th><th>LOG</th></tr>
<tr><td>21</td><td>begin checkpoint
Transaction Table:
<table>
<tr><th>XACT</th><th>LastLSN</th></tr>
<tr><td>T1</td><td>12</td></tr>
</table>
Dirty Page Table:
<table>
<tr><th>Page</th><th>recLSN</th></tr>
<tr><td>P2</td><td>18</td></tr>
</table></td></tr>
<tr><td>22</td><td>end checkpoint</td></tr>
<tr><td>23</td><td>UPDATE: T3 writes P1</td></tr>
<tr><td>24</td><td>UPDATE: T2 writes P4</td></tr>
<tr><td>25</td><td>UPDATE: T2 writes P2</td></tr>
<tr><td>26</td><td>COMMIT : T1</td></tr>
<tr><td>28</td><td>END : T1</td></tr>
<tr><td></td><td>xxx CRASH xxx</td></tr>
</table>

(a) (7 points) Draw the Transaction Table and Dirty Page Table as they appear at the *end* of the ANALYSIS phase of recovery.

(b) (3 points) Identify the "loser" transactions, if any.
Answer: _____

(c) (3 points) Where (at which LSN) does the REDO phase of recovery begin?
Answer: _____

(d) (3 points) How many compensation log records (CLRs) are written during the UNDO phase of recovery, if any?
Answer: _____

(e) (4 points) What is the difference between a *system crash* and a *media failure*?