Database Concepts- Assignment 8

1. (a) Consider the following parameters:

block size	=	4096 bytes
block-address size	=	9 bytes
block access time	=	10 ms (micro seconds)
record size	=	200 bytes
record key size	=	12 bytes

Assume that there is a B^+-tree, adhering to these parameters, that indexes 100 million records on their primary key values.

i. Specify (in ms) the minimum time to retrieve a record with key k in the B^+-tree provided that there is a record with this key.

ii. Specify (in ms) the maximum time to retrieve a record with key k in the B^+-tree.

iii. How many records would there need to be indexed to increase the maximum time to retrieve a record with key k in the B^+-tree by at least 20 ms?

iv. How would your answer to question 1(a)ii change if the block size is 8192 bytes.

Show all the intermediate computations leading to your answers.

(b) Consider the following B^+-tree of order 2 that holds records with keys 2, 7, 9, and 11. (Observe that (a) an internal node of a B^+-tree of order 2 can have either 1 or 2 keys values, and 2 or 3 sub-trees, and (b) a leaf node can have either 1 or 2 key values.)

```
           -----------
           | /  9  \ |
           -----------
           /         \
          /           \
 ----------     ------------
 | 2   7 |--->| 9      11 |
 ----------     ------------
```

i. Show the contents of your B^+-tree after inserting records with keys 6, 10, 14, and 4, in that order.

ii. Starting from your answer in question 1(b)i, show the contents of your B^{+}-tree after deleting records with keys 2, 14, 4, and 10, in that order.

2. (a) Consider an extensible hashing data structure wherein (1) the initial global depth is set at 1 and (2) all directory pointers point to the same **empty** block which has local depth 0. So the hashing structure looks like this:

```
global-depth 1       local-depth 0
     ------         ------------------
   0 |  --|-------->|                |
     ------         |                |
   1 |  --|-------->|                |
     ------         ------------------
```

Assume that a block can hold at most two records.

i. Show the state of the hash data structure after each of the following insert sequences:[1]

A. records with keys 2 and 6.

B. records with keys 1 and 7.

C. records with keys 4 and 8.

D. records with keys 0 and 9.

ii. Starting from the answer you obtained for Question 2(a)i, show the state of the hash data structure after each of the following delete sequences:

A. records with keys 1 and 2.

B. records with keys 6 and 7.

C. records with keys 0 and 9.

[1] You should interpret the key values as bit strings of length 4. So for example, key value 7 is represented as the bit string 0111 and key value 2 is represented as the bit string 0010.

3. Let $R(A, B)$ and $S(B, C)$ be two relations and consider their natural join $R \bowtie S$.

   Assume that R has 1500000 records and that S has 5000 records.

   Furthermore, assume that 30 records of R can fit in a block and that 10 records of S can fit in a block.

   Assume that you have a main-memory buffer with 101 blocks. (One of these blocks is reserved for output purposes.)

   (a) How many block IO's are necessary to perform $R \bowtie S$ using the nested-loops join algorithm? Show your analysis.

   (b) How many block IO's are necessary to perform $R \bowtie S$ using the merge-join algorithm? Show your analysis.

   (c) How many block IO's are necessary to perform $R \bowtie S$ using the hash-join algorithm? Show your analysis.

   Note: For questions 3b and 3c you can assume that the buffer is sufficiently large to hold all records from R and S for any give B-value.

4. State which of the following schedules S1, S2, and S3 over transactions T1, T2, and T3 are conflict-serializable, and for each of the schedules that is serializable, given a serial schedule with which that schedule is conflict-equivalent.

    (a) S1 = R1(x) R2(y) R1(z) R2(x) R1(y).

    (b) S2 = R1(x) W2(y) R1(z) R3(z) W2(x) R1(y).

    (c) S3 = R1(z) W2(x) R2(z) R2(y) W1(x) W3(z) W1(y) R3(x).

5. Give 3 transactions T_1, T_2, T_3 and a serializable schedule S on these transactions whose precedence graph (i.e. serialization graph) consists of the edges (T_1, T_2) and (T_1, T_3). Give 2 serial schedules that are conflict-equivalent with S.

6. Give 3 transactions T_1, T_2, T_3 and a schedule S on these transactions whose precedence graph (i.e. serialization graph) consists of the edges (T_1, T_2), (T_2, T_3), and (T_3, T_1). Is your schedule S serializable?

7. Give 3 transactions T_1, T_2, and T_3 that each involve read and write operations and a schedule S that is conflict-equivalent with **all** serial schedules over T_1, T_2, and T_3.

8. Consider the two transactions $$T_1\text{:}\ \begin{array}{l} \text{R}(X) \\ \text{W}(X) \\ \text{R}(Y) \\ \text{W}(Y) \end{array} \quad \text{and } T_2\text{:}\ \begin{array}{l} \text{R}(Z) \\ \text{W}(Y) \\ \text{W}(X) \end{array}$$

    (a) List the pairs of conflicting operations in T_1 and T_2.

    (b) Give two schedules involving transactions T_1 and T_2 that are non-serializable. Prove that your schedules are non-serializable.

    (c) Augment the transactions T_1 and T_2 with locking operations so that they satisfy the two-phase locking protocol.

    For these updated transactions specify a deadlock-free, non-serial schedule. Is this schedule serializable? Show why?

    If so what is an equivalent serial schedule?

9. Consider the following transactions:

```
T1:  read(A);
     read(B);
     if A = 0 then B := B+1;
     write(B).

T2:  read(B);
     read(A);
     if B = 0 then A := A+1;
     write(A).
```

Let the consistency requirement be $\mathtt{A} = 0 \vee \mathtt{B} = 0$, and let $\mathtt{A} = \mathtt{B} = 0$ be the initial values.

(a) Show that each serial schedule involving transaction T1 and T2 preserves the consistency requirement of the database.

(b) Construct a schedule on T1 and T2 that produces a non-serializable schedule.

(c) Is there a non-serial schedule on T1 and T2 that produces a serializable schedule. If so, give an example.