CS43002: Database Management Systems

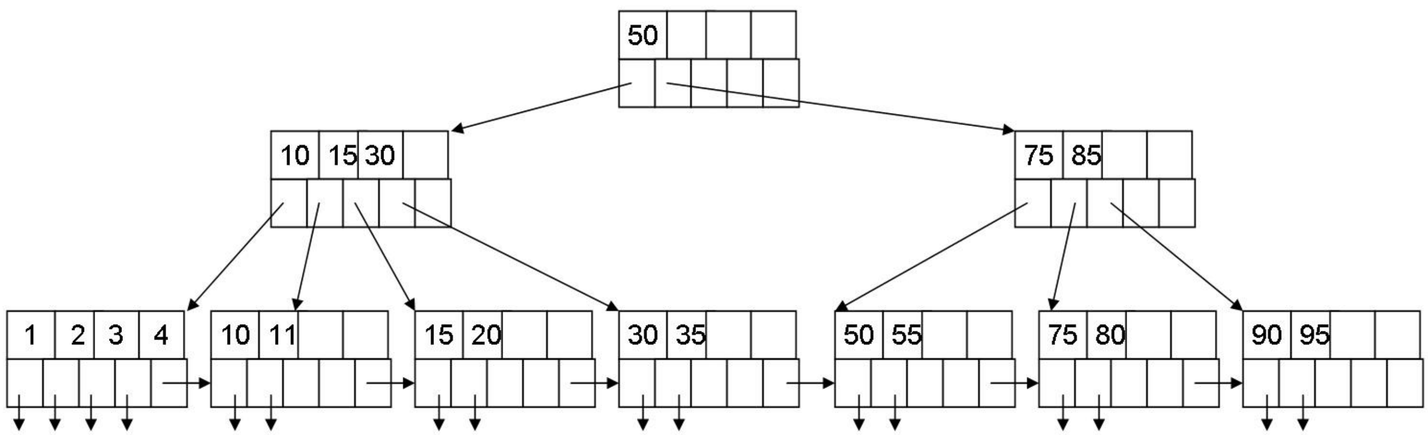
End-semester Examination, Spring 2016

Time= 3 hrs. Marks: 5x12 = 60. Answer all five questions.

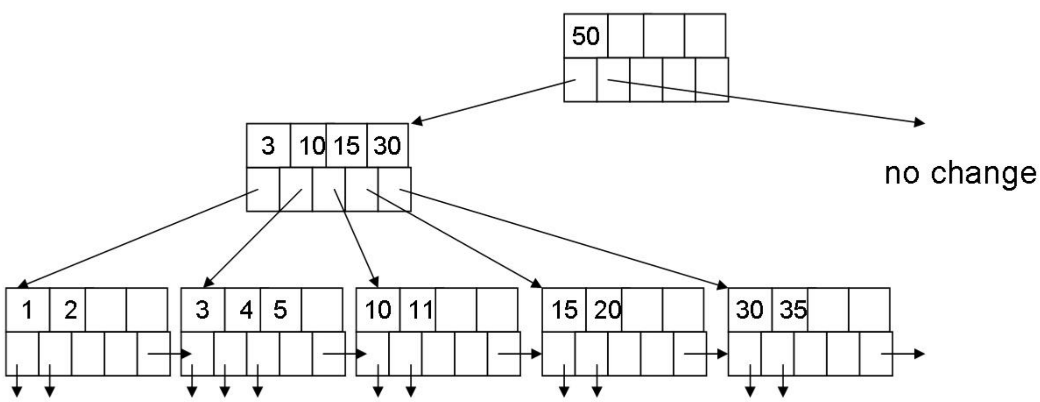
Answer all parts of a single question together.

---

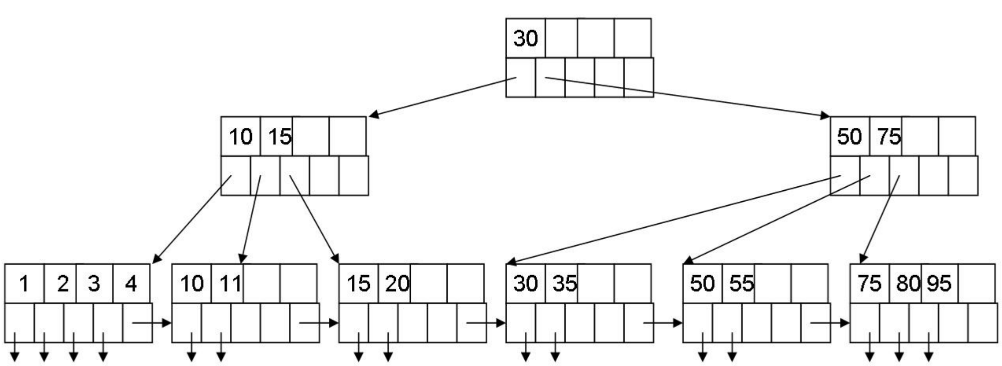
1. Consider the following B+ tree of order 4 (i.e., each node can hold at most 4 keys and 5 pointers).



(a) Show the resulting tree after inserting key 5 in the original tree. [4]



(b) Show the resulting tree after deleting key 90 in the original tree. [4]



(d) Describe how variable-length records are stored in a block. [4]

2. (a) Consider two relations R(a, b, c, d) and S(d, e) with the following statistics: [4]

T(R) = 100; V(R, a) = 100; V(R, b) = 10; V(R, c) = 1; V(R, d) = 50; T(S) = 500; V(S, d) = 30; V(S, e) = 100;

Estimate the number of tuples in: i. $\sigma_{b = 25}$(R), ii. $\sigma_{(b = 25) \wedge (c = 30)}$(R), iii. $\sigma_{(b = 25) \vee (d = 30)}$(R), iv. R $\bowtie$ S

Ans:
i. T(R)/V(R, b) = 10 ii. T(R)/(V(R, b)V(R, c)) = 10 iii. T(R)/V(R,b) + T(R)/V(R.c) – T(R)/(V(R,b)V(R,c)) = 12
iv. T(R)T(S)/max(V(R,d)V(S,d)) = 1000

(b) Consider the following query: [4]

SELECT *
FROM R, S, T
WHERE R.A = S.A and R.B = T.B AND T.C = S.C
AND R.D = 16 AND S.F = 17

Show "two" relational algebra expression trees for this plan, that : (I) contain no cartesian product, (II) have pushed selections down, (III) each join is associated with an atomic condition (not a conjunctive condition).

(c) We have a relation whose *n* tuples each require *R* bytes, and we have a machine whose main memory *M* bytes and disk block size *B* bytes are just sufficient to sort *n* tuples using *two*-phase multiway merge sort. Now we are given an integer *k*. How many records can be sorted using *k*-phase multiway merge sort? [4]

Ans: The number of sorted sublists s we need is s = nR/M. On the second phase we need one block for each sublist, plus one for the merged list. Thus, we require Bs < M. Substituting for s, we get nRB/M < M, or n < M^2*/RB.*

3. Consider the following two schedules S1 and S2. The actions are listed in the order they are scheduled and prefixed with the transaction name. [4 x 3]

S1 = T1:W(X); T2:R(X); T1:W(X); T2:Abort; T1:Commit

S2 = T2:R(X); T3:W(X); T3:Commit; T1:W(Y); T1:Commit; T2:R(Y); T2:W(Z); T2:Commit

(a) For each of the above schedules – draw the precedence graph, and decide which of the following four classes the schedules belongs to. Classes are: conflict-serializable, view-serializable, recoverable, avoids cascading roll-back.

Ans: S1: Not conflict serializable, not view serializable, not recoverable, not avoiding cascading rollback

S2: Belongs to all the four classes.

(b) Consider the following schedule S. There are three transactions, each reading A and writing B.

S = T1:R(A); T1:W(B); T2:R(A); T2:W(B); T3:R(A); T3:W(B);

How many schedules involving the six actions in S are conflict equivalent to S?

Ans: 15. The three writes have to appear in the order shown, so let's start with the sequence w1(B), w2(B), w3(B), and see where the other three actions could fit in. First, r1(A) must appear before w1(B) because we cannot reorder actions of a transaction. Thus, there is only one choice so far: r1(A), w1(B), w2(B), w3(B). Now, r2(A) can appear in any of the points before w2(B), so there are three choices of where to put r2(A). Finally, *regardless of where we put r2(A), there are five choices of where to put r3(A) --- anywhere ahead of w3(B). Thus, there are 15 possible schedules.*

(c) Show that if two schedules are conflict equivalent, then they are view equivalent.

4. (a) For the schedules S1 and S2 shown in Question 3, explain which of the following protocols allows it. Assume that transaction Ti, has timestamp i. [6 + 6]

2PL, Strict 2PL, Timestamp without Thomas write rule, Timestamp with Thomas write rule.

Ans: S1: Not 2PL, Not S2PL, Not Timestamp without Thomas rule, Not Timestamp with write rule.

S2: Not 2PL, Not S2PL, Timestamp without Thomas rule, Timestamp with write rule.

(b) Suppose that we run the transactions T1-T6 using the validation protocol. The following table lists their read and write sets. No other transactions exist.

Transaction	Read	Write
T1	{a, b}	{b, c}
T2	{a, b, c}	{h}
T3	{b}	{d, e}
T4	{c}	{f, g}
T5	{a}	{d, f}
T6	{g}	{e, g}

The following sequence of events takes place. Explain the outcome (commit/abort) of each transaction T1-T6.

1. T1, T2, T3, T4 starts (in this order)

2. T3 initiates validation

3. T5, T6 starts (in this order)

4. T1 initiates validation

5. T5 initiates validation

6. T4 initiates validation

7. T2 initiates validation

8. T1, T2, T3 finish (if they are not aborted earlier)

9. T6 initiates validation

10. T4, T5, T6 finish (if they are not aborted earlier)

Ans: T1: C, T2: A, T3: C, T4: A, T5: A, T6: C

5. (a) Describe two strategies for deadlock prevention in a concurrency control system of a database. [3]

i. Wait-Die

ii. Wound-wait

(b) List the steps followed while checkpointing in a log based recovery system. [3]

i. Output to stable storage all log records currently in main memory

ii. Output to the disk all modified buffer blocks

iii. Outputs to stable storage a log record <checkpoint, L> where L is the list of all active transactions

(c) Suppose A and B are database elements and their initial values are both 0. After transaction T executes their values are both changed to 1. The contents of the log are given below: [3]

<START T> <T, A, 0> <T, ?, ?> <COMMIT T>

What kind of logging was used, and what are the values for the two question marks?

Ans: Undo and B, 0

(d) Consider the following Undo/Redo log: [3]

<START T1> <START T2> <START T3> <T1, A, 0,1 > <T2, A, 1, 2> <T3, A, 2, 3> <COMMIT T2>

If we follow undo/redo recovery and ignore needs for cascaded rollbacks, then the value of A on disk after recovery is?

Ans: 2

------------ BEST WISHES ------------