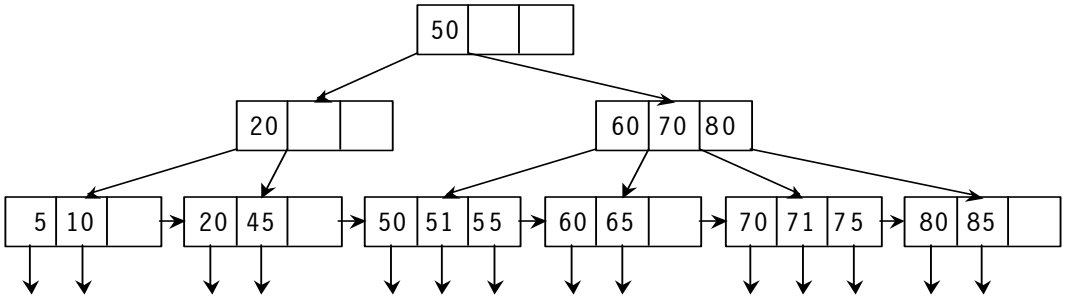
CPS 116 Fall 2004
Homework #4 (8.75% of course grade: 100 points)
Assigned: Thursday, November 18
Due: Thursday, December 2

Problem 1 (20 points).

For each of the following modifications, show the result B^+-tree obtained by applying the modification to the B^+-tree shown below. Suppose that the maximum fan-out is 4. (Always start with the B^+-tree shown below; do not apply the modifications to the result of previous modifications.)



(a) Insert 21.
(b) Delete 50.
(c) Insert 79.
(d) Delete 10.

Problem 2 (12 points).

A table $R(\underline{K}, A, \ldots)$ with 100,000 rows is stored in 10,000 disk blocks. The rows are sorted by K, but not by A. There is a dense, secondary B^+-tree index on $R(A)$, which has 3 levels and 500 leaves.

Suppose we want to sort R by A. We have 101 memory blocks at our disposal. Method 1 performs an external-memory merge sort using all memory available. Method 2 takes advantage of the fact that the values of A are already sorted in the B^+-tree index on $R(A)$: It simply scans the leaves of the index to retrieve and output R rows in order.

How many disk I/O's do these two methods require? Which one is the winner?

Problem 3 (35 points).

Consider tables $R(A, B, C)$, $S(C, D)$, and $T(D, E)$. Transform the following query into an equivalent query that:

- Contains no cross products;
- Performs projections and selections as early as possible.

(a) $\pi_{R.B,\, S.D,\, T.E}\, \sigma_{(R.A=10) \text{ and } (R.C = S.C) \text{ and } (S.D = T.D) \text{ and } (R.A > T.E)}\, (R \times S \times T)$

Suppose we have the following statistics:

- $| R | = 1{,}000$; $| \pi_A R | = 1{,}000$; $| \pi_B R | = 100$; $| \pi_C R | = 500$;
- $| S | = 5{,}000$; $| \pi_C S | = 300$; $| \pi_D S | = 10$;
- $| T | = 4{,}000$; $| \pi_D T | = 4{,}000$; $| \pi_E T | = 1{,}500$.

Estimate the number of the tuples returned by the following queries:

(b) $\sigma_{A=10}\, R$
(c) $\sigma_{A=10 \text{ and } B = \text{“Bart”}}\, R$
(d) $\sigma_{A=10 \text{ or } B = \text{“Bart”}}\, R$
(e) $R \bowtie S$
(f) $R \bowtie S \bowtie T$

For the following question, further suppose that:

- Each disk/memory block can hold up to 10 tuples;
- All tables are stored compactly on disk (10 tuples per block) in no particular order;
- No indexes are available;
- 11 memory blocks are available for query processing.

(g) What is the best execution plan (in terms of number of I/O's performed) you can come up with for the query $\sigma_{R.B = \text{“Bart”} \text{ and } S.D = 100}\, (R \bowtie S)$? Describe your plan and show the calculation of its I/O cost.

Problem 4 (15 points).

For each schedule below, tell whether it is conflict-serializable. If yes, also tell:

- Whether it is recoverable;
- Whether it avoids cascading rollbacks;
- Whether it is possible under *strict* 2PL.

(a) T_1.write(B), T_2.read(A), T_2.write(A), T_1.read(A), T_1.write(A), T_1.commit, T_2.commit
(b) T_1.write(B), T_2.read(A), T_2.write(A), T_1.read(A), T_1.write(A), T_2.commit, T_1.commit
(c) T_1.write(B), T_2.read(A), T_2.write(A), T_2.commit, T_1.read(A), T_1.write(A), T_1.commit
(d) T_1.write(B), T_2.read(A), T_1.read(A), T_2.write(A), T_1.write(A), T_2.commit, T_1.commit
(e) T_2.write(B), T_2.read(A), T_2.write(A), T_1.write(B), T_2.commit, T_1.read(A), T_1.commit

Problem 5 (18 points).

Consider the following transaction log from the start of the run of a database system that uses undo/redo logging with fuzzy checkpointing:

1. ⟨ T1, start ⟩
2. ⟨ T1, *A*, 45, 10 ⟩
3. ⟨ T2, start ⟩
4. ⟨ T2, *B*, 5, 15 ⟩
5. ⟨ T2, *C*, 35, 10 ⟩
6. ⟨ T1, *D*, 15, 5 ⟩
7. ⟨ T1, commit ⟩
8. ⟨ T3, start ⟩
9. ⟨ T3, *A*, 10, 15 ⟩
10. ⟨ begin-checkpoint {T2, T3} ⟩
11. ⟨ T2, *D*, 5, 20 ⟩
12. ⟨ T2, commit ⟩
13. ⟨ end-checkpoint ⟩
14. ⟨ T4, start ⟩
15. ⟨ T4, *D*, 20, 30 ⟩
16. ⟨ T3, *C*, 10, 15 ⟩
17. ⟨ T3, commit ⟩
18. ⟨ T4, commit ⟩

What is the value of the data items *A, B, C,* and *D* on disk after recovery:

(a) if the system crashes just before line 6 is written to disk?
(b) if the system crashes just before line 10 is written to disk?
(c) if the system crashes just before line 12 is written to disk?
(d) if the system crashes just before line 13 is written to disk?
(e) if the system crashes just before line 16 is written to disk?
(f) if the system crashes just before line 18 is written to disk?