CS245 Database System Principles

Midterm Exam Winter 2001

This exam is open book and notes. You have 70 minutes to complete it.

Print your name: ____

The Honor Code is an undertaking of the students, individually and collectively:

1. that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;
2. that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.

The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.
While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

I acknowledge and accept the Honor Code.
Signed: ____

Problem	Points	Maximum
1		10
2		10
3		10
4		15
5		10
6		15
Total		**70**

Problem 1 (10 points)

State if the following statements are TRUE or FALSE. Write down the answer inside the box on the left

Answer:

a) Insertion order into a B+ tree will effect the tree's end structure.

b) B+ trees are typically more efficient than linear or extensible hashing for all types of queries.

c) The time for a request to wait to be serviced may be longer using the elevator algorithm than using the first-come-first-serve algorithm.

d) Consider relations *R(A,B)* and *S(B,C)* where *T(R) = 5000*, *T(S) = 3000*, and *B* is a primary key on *S*. The expected number of tuples in $R \bowtie S$ is less than or equal to *3000*.

e) Consider two relations *R(A, B, C)* and *S(A, D, E)*, sharing a common attribute *A*. It is known that *R.A* is a foreign key referencing *S.A*, and that *S.A* is the primary key of relation *S*. Then the estimated size of $R \bowtie S$ is *T(R)*.

f) For any data file, it is possible to construct two separate sparse first level indexes on different keys.

g) For any data file, it is possible to construct two separate dense first level indexes on different keys.

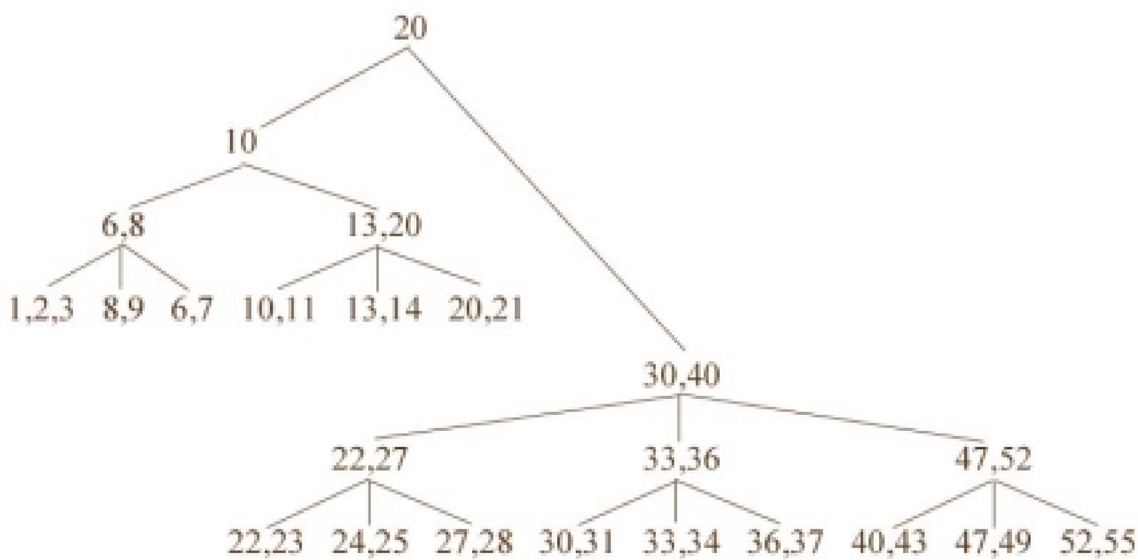
h) For any data file, it is possible to construct a sparse first (lower) level index and a dense second (higher) level index. Both indices should be useful.

i) For any data file, it is possible to construct a dense first (lower) level index and a sparse second (higher) level index. Both indices should be useful.

j) $\pi_S(E_1 - E_2) = \pi_S(E_1) - \pi_S(E_2)$

Problem 2 (10 points)

Find any/all violations of a B+ tree structure in the following diagram. Circle each bad node and give a *brief* explanation of each error. Assume the order of the tree is 4 (*n=4; 4 keys, 5 pointers*).



Answer:

Problem 3 (10 points)

A database system uses a variant of B-trees to maintain sorted tables. In this variant, the leaves contain the records themselves, not pointers to the records. A leaf can contain as many records as will fit in a block (allowing for a sequence pointer). Non-leaf nodes are also one block in size, and contain as many keys (and appropriate pointers) as will fit. Assume the following:

1. Blocks are *4096 bytes*.
2. Each record is *300 bytes* long.
3. A block pointer is *10 bytes*.
4. A record pointer is *12 bytes*.
5. A key for the index is *8 bytes* long.
6. The nodes are *85%* occupied. For example, if a leaf can hold *100* records, it will only hold *85*. If a non-leaf can hold 100 keys, it will only hold *85*. (For 85% calculations, round to the nearest integer.)
7. The indexed file has *1,000,000 records*.

How many blocks will this index use? (Note that the leaves are a part of this index.)

Answer:

Problem 4 (15 points)

For this problem, consider hash indexes with the following characteristics:

1. A block can hold 50 value/pointer pairs. (The hash structure does not contain the records themselves, only pointers to them.)
2. The file being indexed has 1000 records.
3. The indexed field can contain duplicates.

a) For a linear hash index with an average occupancy (utilization) of *50%*, how many blocks will the buckets require in the worst-case?

b) In case (a), what is the worst-case number of I/Os to look up a value (including the record itself)?

c) For an extensible hash index, what is the worst-case (largest possible) size of the directory? If there is no limit, state so.

d) For an extensible hash index, what is the best-case (smallest possible) size of the directory?

Answer:

Problem 5 (10 points)

Parts *a, b* and *c* below refer to disk with an actual (formatted) capacity of *8 gigabytes* (2^{33} *bytes*). The disk has *16 surfaces* and *1024 tracks*. The disk rotates at *7200 rpm*. The average seek time is *9 ms*. The block size is *8KB*.

For each of the following questions, state if there is enough information to answer. If there is, give the answer. If not, explain what is missing.

a) What is the capacity of a single track?

b) Suppose we are reading a file F that occupies exactly one entire track. Assume that a track can only be read starting at a particular position on the track. How long does it take to read the entire file sequentially?

c) How long does it take to read a block?

Answer:

Problem 6 (15 points)

Consider a relation *R(A,B)* stored using a partitioned hash organization. The hash table contains *1024 blocks* (buckets), and the tuples (records) are stored in these blocks.

To store tuple *(a,b)*, we apply hash function *h1* for *a*, obtaining *X* bits. Then we apply hash function *h2* on *b*, obtaining *10-X* bits. The bits are concatenated, obtaining a 10 bit hash that identifies the block where *(a,b)* is placed..

Suppose that *20%* of the queries on R are of the form Q_1: `SELECT * from R` where *A=a*, and *80%* of the queries are of the form Q_2: `SELECT * from R where B=b` where *a* and *b* are constants.

a) How many blocks are accessed by the Q_1 queries? How many by the Q_2 quereis? (your answer will be a function of *X*.)

b) Write an expression that gives the expected number of blocks that must be accessed (on average).

c) What *X* value minimizes the expected number of IOs? (Hint: Recall that the derivative (with respect to *z*) of 2^{az+b} is $a2^{az+b}ln2$.)

Answer: