Rasmus Pagh
IT University of Copenhagen

Advanced Database Technology

April 24, 2006

COURSE OVERVIEW

Next

- An overview of the material of the course, based on various kinds of questions one could imagine at the exam.
- NB! The exam questions have not yet been written, so you won't find any hints or anti-hints :-)
- An overview of some DB/algorithms related thesis topics.

I/O model

- How many I/Os are needed if the following entries in an external memory array are to be read: ...?
- How many I/Os are used by the following algorithm ... if LRU caching is used? Can you improve the number of I/Os by a better caching strategy?

Basic I/O algorithms

- Describe an I/O efficient algorithm for the following problem: ... (Hint: Reduce the problem to sorting/scanning).
- How much internal memory is needed for 2-pass merge sort in the following setting: ...
- Argue that the following problem requires Sort(n) I/Os. (Hint: Show how to sort by solving an instance of the problem).

Representing data elements

- Write a short description of the following ways of dealing with record modifications, pointing out any disadvantages and ways of addressing them: 1) Overflow blocks, 2) Tombstones.
- Consider the following linked list. Suppose we insert keys as follows: ... using the method described in [Pagh03]. What will be the resulting data structure?
- What is the amortized cost per operation in the following data structure: ...

B-tree and hash indexes

- What is the result of inserting/deleting the following keys in the B-tree/hash table shown: ...?
- What is the cost of computing the following relational algebra expression: ...
  1) Without index. 2) With B-tree index.
- Suggest an index that could help in the following query, and explain why: ...?
- Analyze the time per operation of the following variant of a B-tree/hash index.

Relational operations

- How many I/Os are needed to compute the following relational algebra expression: ...?
- Evaluate the advantages and disadvantages of using duplicate elimination on the intermediate result in the following relational algebra expression: ...
- Analyze the following variant of the algorithm for join described in GUW: ...

Query compilation/optimization

- Use algebraic laws to rewrite the following expression into something simpler: ...
- Estimate the size of all subexpressions in the following expression, using the rules in GUW/using sampling: ...
- Choose the best order of operations, based on the following estimated sizes of subexpressions: ...

System and media failures

- Suppose there is a system crash and the state of the undo/redo log is as follows: ... How will the recovery proceed?
- What kind of disk system would you suggest to protect against the following kinds of failures: ... (The redundancy should be as small as possible.)

Concurrency control

- Are the following schedules conflict-serializable: ... ?
- Consider the following transactions: ...
  - Explain how using 2-phase locking for concurrency control could result in a deadlock if they are run concurrently.
  - Explain how using timestamping for concurrency control could result in one of the transactions being rolled back.

Geometric indexes

- Suppose you have a set of points in a 2-dimensional plane, represented using an R-tree/grid file/persistent B-tree. How could the following query be answered efficiently: ...?
- Devise an efficient index for answering the following kind of query: ... (Hint: Use a persistent B-tree/point location data structure.)

Text indexes

- Draw the Patricia trie/Short Pat Array for the following strings: ...
- In the following String B-tree ... how will a search for ”Mads Tofte” proceed?
- Devise an efficient text index for answering the following kind of query: ...

Data mining

- Which pairs of items will the A-Priori algorithm store in internal memory given the following input baskets: ...?

Thesis/project topics

- Five DB related possibilities on web page (`www.itu.dk/cla/tiki-index.php?page=ProjectProposals`):
  - **Computing joins in databases**
  - **Buffered indexes**
  - **Filtering - with applications in databases (distr.)**
  - **Adaptive sorting & adaptive join processing**
  - **Using multiple disks efficiently**
- Feel free to suggest your own variant - or something completely different.