Lecture 29: Final Review
Wednesday, December 11, 2002

Counting the Number of Join Orders (Exercise)

\[ R_0(A_0,A_1) \ R_1(A_1,A_2) \ldots \ R_n(A_n,A_{n+1}) \]
- The number of left linear join trees is: \( n! \)
- The number of left linear join trees without cartesian products is:
  \[ 2^n \] (why ?)
- The number of bushy join trees is:
  \[ n!(n+1)!C_{2n}^n = (2n)!!/(n+1)!/(n!) \]
- The number of bushy join trees without cartesian product is:
  \[ 2^{n+1}/(n+1)!C_{2n}^n \] (why ?)

Number of Subplans Inspected by Dynamic Programming

\[ R_0(A_0,A_1) \ R_1(A_1,A_2) \ldots \ R_n(A_n,A_{n+1}) \]
- The number of left linear subplans inspected is:
  \[ \sum_{k=1}^{n} C_{n}^{k} k = n2^{n-1} \]
- The number of left linear subplans without cartesian products inspected:
  \[ \sum_{k=1}^{n} (n-k+1)^2 = n(n+1) \] why ?
- The number of bushy join subplans inspected is:
  \[ \sum_{k=1}^{n} C_{n}^{k} 2^k = 3^n \] why ?
- The number of bushy join subplans without cartesian product:
  \[ \sum_{k=1}^{n} (n-k+1)(k-1) = n^2(n-1)/2 - n(n-1)(2n-1)/6 = n(n-1)(n+1)/6 \]

The Final
- Date: Friday, December 13
- Time: 8:30 – 10:20
- Place: this room
- Open book exam!

What to Prepare for the Final
Everything!
- Data modeling
- Relational model
- XML
- Relational algebra
- SQL
- Storage
- Indexes
- Physical operators
- Optimization
- Recovery
<table>
<thead>
<tr>
<th>Data Modeling</th>
<th>Relational Algebra</th>
</tr>
</thead>
<tbody>
<tr>
<td>• E/R diagrams, ODL</td>
<td>• Relations</td>
</tr>
<tr>
<td>• Keys</td>
<td>• Keys</td>
</tr>
<tr>
<td>• Relationships</td>
<td>• Functional dependencies</td>
</tr>
<tr>
<td>• Inheritance</td>
<td>• Decomposition</td>
</tr>
<tr>
<td>• Mapping to relations</td>
<td>• Normal forms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>XML</th>
<th>SQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>• XML syntax</td>
<td>• Select-from-where</td>
</tr>
<tr>
<td>• DTD</td>
<td>• Subqueries</td>
</tr>
<tr>
<td>• From relations to XML</td>
<td>• Aggregation</td>
</tr>
<tr>
<td>• From XML to relations</td>
<td>• Nulls</td>
</tr>
<tr>
<td></td>
<td>• Outer joins</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SQL (continued)</th>
<th>Data Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Database modification</td>
<td>• The I/O model of computation</td>
</tr>
<tr>
<td>• Defining and modifying relation</td>
<td>• Representing data elements:</td>
</tr>
<tr>
<td>schemas</td>
<td>– Grouping records into blocks</td>
</tr>
<tr>
<td>• Constraints</td>
<td>– Variable length records</td>
</tr>
<tr>
<td>– On attribute values</td>
<td>– Overflow blocks</td>
</tr>
<tr>
<td>– Keys</td>
<td></td>
</tr>
<tr>
<td>– Foreign keys</td>
<td></td>
</tr>
<tr>
<td>• Embedded SQL</td>
<td></td>
</tr>
</tbody>
</table>
Index Structures

- Terminology:
  - Dense/sparse index
  - Primary/secondary index
- B*-trees
- Hash tables

Physical Operators

- One-pass algorithms
- Nested-loop joins
- Two-pass algorithms based on sorting
- Two-pass algorithms based on hash tables
- Index-based algorithms

Optimizations

- Extended logical operators
- Algebraic identities
- Heuristic based optimization:
  - Size estimation
  - Dynamic programming for join order

Recovery

- Undo logging
- Redo logging
- Undo/redo logging

General Advice

- Some problems will require thinking
  - Use judgment
- Problem difficulty may be uneven:
  - do the easy ones first

Grading

- Homework 25%
  - best 4 out of 5
- Project: 25%
- Midterm: 20%
- Final: 25%
- Intangibles: 5%
The End