CSE 444 final (second midterm) review

Some possible topics

Database tuning
- B+ trees
- Index selection

Query execution and optimization
- Relational algebra
- Physical query operators
- Estimating statistics
- Selecting logical and physical plans

Parallel databases
- Parallel, distributed query operators
- MapReduce
- Pig Latin

XML
- Syntax of a well-formed XML document
- DTDs
- XPath
- XQuery

Note: this is not a complete list of topics

[Winter 2005 final, problem 2]

Consider an XML document containing information about job postings and job applications. The postings are grouped by company, and applications are listed under postings. An application contains only the applicant's name. For example:

```
<jobs>
  <company>
    <name> MicroScience Corp. </name>
    <posting>
      <jobtitle> sales rep </jobtitle>
      <salary> 30000 </salary>
      <application> Mark </application>
      <application> Lucy </application>
      <application> Frank </application>
    </posting>
    <posting>
      <jobtitle> technology consultant </jobtitle>
      <salary> 80000 </salary>
      <application> Lucy </application>
      <application> Fred </application>
      <application> Mark </application>
      <application> Betty </application>
    </posting>
  </company>
  <company>
    <name> MacroConsulting Inc. </name>
    <posting>
      <jobtitle> technology consultant </jobtitle>
      <salary> 40000 </salary>
      <application> Frank </application>
    </posting>
    <posting>
      <jobtitle> debugger </jobtitle>
      <salary> 20000 </salary>
    </posting>
    <posting>
      <jobtitle> programmer analyst </jobtitle>
      <salary> 35000 </salary>
      <application> Lucy </application>
      <application> Mark </application>
    </posting>
  </company>
</jobs>
```

a. For each of the XPath expressions, indicate how many answers it will return on the example XML document on previous page. For example, if the XPath expression is

/jobs/company[name/text()='MacroConsulting Inc.']/posting

then you will answer 3. For each question, you have to turn in a number.

i. //application

ii. /jobs/company/posting[salary/text()>60000]//application

iii. /jobs/company[posting/salary/text()>60000]//application

b. Write an XQuery expression that returns the names of all applicants who submitted applications to at least two companies. Your query should a list of **<name>** elements, and each element should be included only once. For example, if your query were to run on the XML document shown above, then it will return:

<name> Mark </name>
<name> Lucy </name>
<name> Frank </name>

Recall the following pseudo code for counting words in a document.

```
map(String key, String value):
        // key: document name
        // value: document contents
        for each word w in value:
                EmitIntermediate(w, "1"):

reduce(String key, Iterator values):
        // key: a word
        // values: a list of counts
        int result = 0;
        for each v in values:
                result += ParseInt(v);
        Emit(AsString(result));
```

Working with R(A, B) S(C,D), implement the following in Map Reduce pseudo code:

a) Select * from R where A<9 [selection]

b) Select distinct A from R [duplicate elimination]

c) Select * from R, S where R.A = S.C [join]

(a) **[8 points]** Consider two tables $R(A, B)$ and $S(C, D)$ with the following statistics:

$$\begin{aligned} B(R) &= 5 \\ T(R) &= 200 \\ V(R, A) &= 10 \\ B(S) &= 100 \\ T(S) &= 400 \\ V(S, C) &= 50 \\ M &= 1000 \end{aligned}$$

There is a clustered index on $S.C$ and an unclustered index on $R.A$. Consider the logical plan:

$$P = \sigma_{A=??}(R) \bowtie_{B=C} S$$

There are two logical operators, $S = \sigma_{A=??}$ and $J = \bowtie_{B=C}$, and for each we consider two physical operators:

$$\begin{aligned} s1 &= \text{one pass table scan} \\ s2 &= \text{index-based selection} \\ j1 &= \text{main memory hash join} \\ j2 &= \text{index-based join} \end{aligned}$$

Both $s1$ and $s2$ are pipelined, i.e. the result of the select operator is not materialized. For each of the resulting four physical plans compute its cost in terms number of disc I/Os, expressed as a function of the statistics above. Your answer should consists of four expressions, e.g. $\texttt{COST}(s1j1) = B(R)B(S)/M + V(R, A)$ (not the real answer).

i. $\texttt{COST}(s1j1) =$

ii. $\texttt{COST}(s2j1) =$

iii. $\texttt{COST}(s1j2) =$

iv. $\texttt{COST}(s2j2) =$

(b) [**2 points**] Indicate the cheapest plan of the four, together with its cost expressed as a number. You will get credit for this point only if you compute correctly all four expressions above.

[Fall 2009 final, problem 2]
We would like to explore logical query plans involving data about downhill skiing races. Three tables are involved:

Athlete(aid, name, country)
Event(eid, year, location)
Result(eid, aid, time)

The Athlete table gives a unique id (aid) for each athlete and his/her name and country. The Event table gives a unique id (eid) for each event and the event year and location (for example, 2010, Vancouver). The Result table gives the athlete's time for each race that he/she participated in. In the Result table, eid and aid are foreign keys in the Athlete and Event tables.

We have the following statistics for these tables:

B(Athlete) = 100	B(Event) = 40	B(Result) = 250
T(Athlete) = 2,000	T(Event) = 400	T(Result) = 50,000
V(Athlete, country) = 100		V(Event, year) = 50
V(Event, location) = 25		

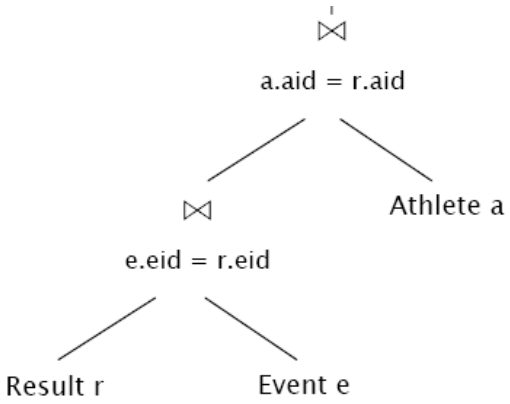
The Athlete and Event tables are clustered on their primary keys, aid and eid respectively. The Result table is not clustered. There are separate indices on all attributes of all tables except for time in Results.

You should assume that there is more than enough main memory (M) to hold any or all parts of these tables so we can use one-pass algorithms to implement queries.

(a) Translate the logical query plan in the diagram on the previous page to SQL.

(b) What is the estimated cost of the logical query plan given in the diagram on the previous page? For full credit you should both give formulas involving things like T(…), B(…), and V(…, …), and then substitute actual numbers and give a numerical answer as your final estimate.

[Fall 2009 final, problem 3]
For this question we go back to the original set of joins at the base of the tree from the previous problem, using the same relations with the same statistics as before.



If we decide to implement this part of the logical plan exactly as given, what physical plan should we use to perform it? Your answer should specify the physical join operators used (hash, nested loop, sort-merge, etc.) and the access methods used to read the tables (sequential scan, index, etc.) For operations where it matters, be sure to include the details – for instance, for a hash join, which relation would be stored in the hash tables; for a loop join, which relation would be the inner or outer loop. You should specify how the top-most join reads the result of the lower one. You may assume there is enough main memory available for whichever algorithm(s) you choose.

Give the estimated cost of your physical plan in terms of the number of disk operations needed. Also give a brief explanation of why your plan is the best one in terms of the overall cost.