Written Assignment #3

Due Date and Time: Thursday, November 17, 2016.

Problem 1. [10 pts]

1. Suppose we wish to sort the following values: 84, 22, 19, 11, 60, 68, 31, 29, 58, 23, 45, 93, 48, 31, 7. Assume that: you have three pages of memory for sorting; you will use an external sorting algorithm with a 2-way merge; a page only holds two values. For each sorting pass, show the contents of all temporary files (like slides 5 in lecture 9).

2. If you have 100 pages of data, and 10 pages of memory, what is the minimum number of passes required to sort the data.

3. If yo have 500,000 pages of data, what is the minimum number of pages of memory required to sort the data in 3 passes.

Problem 2. [10 pts]

Consider the Movie Stars database that contains the following relations:

StarsIn(movieTitle, movieYear, starName)
MovieStar(Name, address, gender, birthdate)

For the following SQL query:

```
SELECT movieTitle
FROM StarsIn, MovieStar
WHERE starName = Name and birthdate LIKE '%1960';
```

write an efficient relational algebra expression that is the equivalent to this query and give an evaluation plan for this expression (choose the algorithm to implement each operation). Justify your answer.

Problem 3. [35 pts]

Consider the join $R \bowtie S$ where the join predicate is $R.a = S.b$, given the following metadata about R and S:

- Relation R contains 2,000 tuples and has 10 tuples per block
- Relation S contains 10,000 tuples and has 10 tuples per block
- Attribute a of relation R is the primary key for R, and every tuple in R matches 5 tuples in S
- There exists a primary index on $S.b$ with height 3
- There exists a secondary index on $R.a$ with height 2
- The buffer can hold 5 blocks

Answer the following questions:

1. If $R \bowtie S$ is evaluated with a block nested loop join, which relation should be the outer relation? Justify your answer. What is the cost of the join in number of I/O's?
2. If $R \bowtie S$ is evaluated with an index nested loop join, what will be the cost of the join in number of I/O's? Show your analysis.
3. What is the cost of a plan that evaluates this query using sort-merge join. Show the details of your cost analysis.
4. If the buffer can hold 202 blocks rather than 5, which of your answers for 1-3 change (if any)? Explain with new cost analyses.
5. Evaluate the cost of computing the $R \bowtie S$ using hash join assuming: i) The buffer can hold 202 blocks, ii) The buffer can hold 5 blocks.

Problem 4. [35 points] Consider the following schema:

Sailors(<u>sid</u>, sname, rating, age) Reserve(<u>sid</u>, <u>did</u>, <u>day</u>) Boats(<u>bid</u>, bname, size)

Reserve.sid is a foreign key to Sailors and Reserves.bid is a foreign key to Boats.bid. We are given the following information about the database: Reserves contains 10,000 records with 40 records per page.
Sailors contains 1000 records with 20 records per page.
Boats contains 100 records with 10 records per page.
There are 50 values for Reserves.bid.
There are 10 values for Sailors.rating(1..10).
There are 10 values for Boat.size
There are 500 values for Reserves.day.

Consider the following query:

```
SELECT S.sid, S.sname, B.bname
FROM Sailors S, Reserves R, Boats B
WHERE S.sid=R.sid AND R.bid = B.bid AND B.size>5 AND
R.day='July 4, 2003'
```

(a) Assuming uniform distribution of values and column independence, estimate the number of tuples returned by this query.

Consider the following query:

```
SELECT S.sid, S.sname, B.bname
FROM Sailors S, Reserves R, Boats B
WHERE S.sid=R.sid AND R.bid = B.bid
```

(b) Draw all possible left-deep query plans for this query.

(c) For the first join in each query plan (the one at the bottom of the tree), what join algorithm would work best? Assume that you have 50 pages of memory. There are no indexes, s o indexed nested loop is not an option. You must consider all possible join algorithms.

Problem 5. [10 points] We discussed using Histogram to estimate number of elements in a query range in class (i.e., selectivity estimation). Another useful technique to achieve similar objective is to use *random sampling*. More specifically, consider a multi-set $X = \{x_1, x_2, \ldots, x_n\}$, where $x_i \in U$ (U is a discrete universe of possible values that elements in X may take); it's possible that $x_i = x_j$ for $i \neq j$.

For any value $a \in U$, we define $f_a(X) = |\{x_j | x_j = a, \text{ and } x_j \in X\}|$. Now, suppose we produce a set S of samples from X with a sampling rate $\rho \in (0, 1)$, i.e., $S = \{s_1, \ldots, s_m\}$ where $m = \rho \cdot n$, and $S \subset X$, and for any $x_i \in X$, $\Pr[x_i \in S] = \rho$; furthermore, these are independent samples, for any $x_i, x_j \in X$, $\Pr[x_i \in S \text{ and } x_j \in S] = \rho^2$. Assume that we use "sampling without replacement", i.e., the same element $x_i \in X$ is never sampled twice into S (but note that different elements with the same value may be both sampled into S).

Use $\mathcal{T}$ to denote all such possible sample sets of S. Prove the followings.

(a) If we build an estimator: $\widehat{f}_{a,S\leftarrow\mathcal{T}}(S) = |\{s_j | s_j = a, \text{ and } s_j \in S\}| \cdot \frac{1}{\rho}$. Prove that $\widehat{f}_{a,S\leftarrow\mathcal{T}}(S)$ is an unbiased estimator for $f_a(X)$.

(b) For any value $a, b \in U$, $a < b$, define $f_{[a,b]}(X) = |\{x_j | a \leq x_j \leq b, \text{ and } x_j \in X\}|$, and $\widehat{f}_{[a,b],S\leftarrow\mathcal{T}}(S) = |\{s_j | a \leq s_j \leq b, \text{ and } s_j \in S\}| \cdot \frac{1}{\rho}$. Prove that $\widehat{f}_{[a,b],S\leftarrow\mathcal{T}}(S)$ is an unbiased estimator of $f_{[a,b]}(X)$.

*** Note that a random variable Y is an unbiased estimator of a value of interest C, iff $\mathbf{E}(Y) = C$; $\mathbf{E}(Y)$ is the expectation of Y.**