Introduction to Database Systems

Problem Set 2
Due: February 25, 2015 at 11:59 PM

1. You have a relation with the values:

   ```
   (47, 1, 6, 12, 21, 14, 23, 26,10,38)
   ```

   Presume they are inserted in this order.

   - Represent this as a B+ tree with 3 keys per node.
   - Draw an extendible hash index for this using the hash function v mod 2^k, where k is the number of bits we consider for the hash. Presume that each page contains 3 values.

2. Start with the B+-Tree from above. Display it after inserting the values 13 and 26, where duplicates are stored. What about if you delete 12? Do these operations in sequence on the same tree. Show the three trees that result.

3. Start with a database having the following tables and page counts:

   Employee - 100 pages
   Office - 30 pages
   Dept - 10 pages

   How many buffer pool pages need to be reserved for the following physical query plans?

   - SeqScan(Office)
   - NestedLoop(SeqScan(Dept),RandomIndexScan(Employee))
   - NestedLoop(NestedLoop(SeqScan(Dept), SeqScan(Office)), RandomIndexScan(Employee))

4. If you have a queue of queries, having page size requirements as follows:
   `(24, 31, 53, 84, 44, 74, 2, 52, 99, 30, 58, 3, 59, 6)`

   And a buffer pool with 100 pages, in what order will these queries be executed? In what batches will they execute? Presume that they all have the same runtime.

5. For each of the following queries, can query rewrite rules be used to simplify them? If not, why not?

   ```
   with raises as
   (SELECT name, salary * 1.1 as new_salary
     FROM employees)
   SELECT employee_name, new_salary
   FROM raises
   WHERE new_salary > 125000;

   with names as (
         SELECT  distinct first_name
         FROM users)
   SELECT count(*) from names;
   ```

```
WITH customer_cnts as (
    SELECT account_id, count(*) as cnt
    FROM account
    GROUP BY account_id)
SELECT account_id
FROM customer_cnts
WHERE cnt > 2 + 3;
```

6. You are joining two tables with the following values:
   A(8,5,7,2,1)
   B(6,9,2,4,5)

   Show the steps for this join using a nested loop, sort-merge, and simple hash join. Presume that the inputs fit in memory, that each input resides on a single page, and that both comparison columns are primary keys. The hash function for the last join is v mod 3.

7. For a database with three tables:

Table	NCARD	Col. Name	ICARD	Min	Max
A	1000	c	91	10	100
B	2000	d	46	5	50
C	5000	e	100	1	100
		f	10	1	10

   - For the query $(\sigma_{c>30}(A)) \bowtie_{c=d} B \bowtie_{d=e} (\sigma_{f\leq 7}(C))$
     Draw out a left-deep query plan for this with the leaf nodes in the order above, and label it with the estimated size of its intermediate results at each step. What is the final estimated output cardinality?
   - For the join $A \bowtie_{c=d} B \bowtie_{d=e} C$, enumerate all of the possible join orders, restricting yourself to left-deep ones. Identify the lowest cost one using the Selinger method. Presume that all joins are sort-merge ones using sequential scans, and that all relations and intermediate results fit into memory. Use 0.001 for w, the CPU cost.