Midterm Exam

CSE232A, Winter 2002

February 21, 2002

Name:

Brief Directions:

- Write clearly: First, you don't want me to spend the whole week grading, do you? Second, it's good for you to write clearly!
- Open books, notes, even databases...
- Good luck!

1 B+ Trees (20 points)

Consider a B+ tree where $n = 4$, i.e., the maximum number of keys in a node is 4 and the maximum number of pointers is 5 at internal nodes and 4 at leaf nodes. Assume that the B+ tree initially consists of a single node, which is both the root and the only leaf, that has the key 1.

1. **2 points** What is the minimum number of keys that may appear in a non-root internal node?

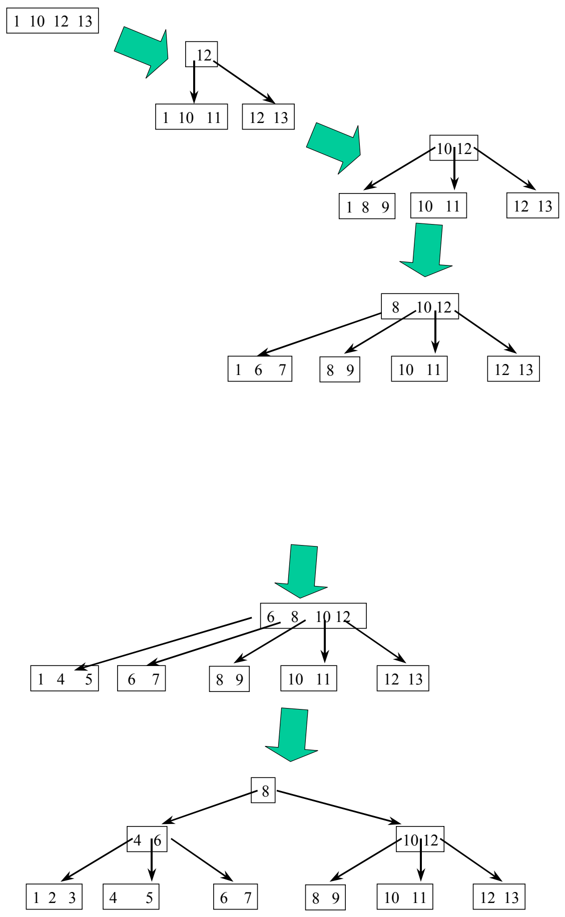
   Solution: $\lceil \frac{n+1}{2} \rceil - 1 = 2$

2. **2 points** What is the minimum number of keys that may appear in a non-root leaf node?

   Solution: $\lfloor \frac{n+1}{2} \rfloor = 2$

3. **8 points**[1] Consider the set of keys $S = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}$. Write down a sequence of inserting the keys of S such that at the end the resulting B+ tree has 3 levels and is as empty as possible, i.e., as many nodes as possible have the minimum number of nodes. Provide the B+ tree snapshots that correspond to the points right after node splits. (Use the white space at the end of the page and the back side of this page.)

   Solution: The sequence $(1), 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2$ produces the sequence of splits and the resulting B+ tree shown below, assuming the splits are "3 nodes to the left, 2 nodes to the right".



   This is just one of the possible solutions. However, the resulting tree is the only one that is possible if the splits are "3 nodes to the left, 2 nodes to the right". There is eactly one possible resulting tree in the case of "2 nodes to the left, 3 nodes to the right".

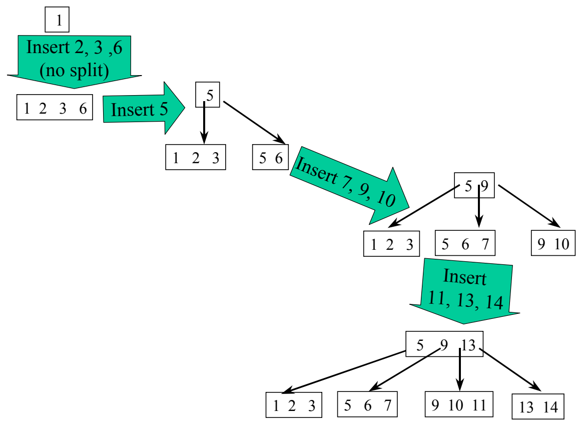
[1]May be time consuming.

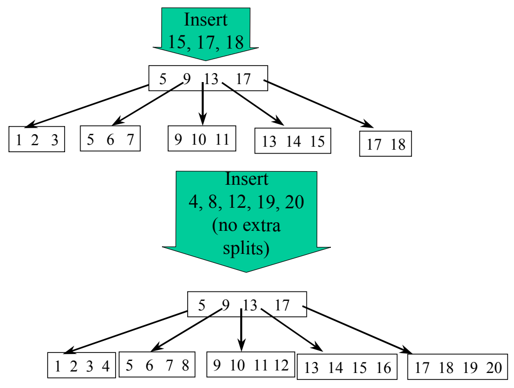
4. **8 points**[2] Consider the set of keys $S' = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20\}$. Write down a sequence of inserting the keys of S' such that at the end the resulting B+ tree is as full as possible, i.e., as many nodes as possible have the maximum number of nodes. Provide the B+ tree snapshots that correspond to the points right after a node split. (Use the next blank page.)

Remarks

- Be consistent in the way you split nodes.
- Assume that there are no duplicate nodes in the internal leaves.

Solution: There is exactly one resulting tree that has just two levels and, hence, is as full as possible. It is produced as follows





[2]May be time consuming

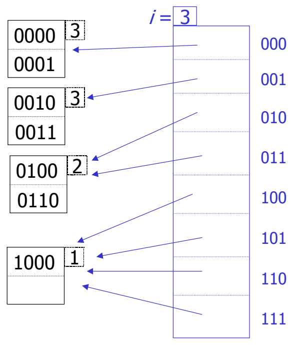
2 Extensible Hash Index (10 points)

Assume that each bucket of an extensible hash index can fit exactly two records (each record is a pair of hash key and pointer). Consider the following records, with the corresponding hash key values.

Key	*hash key*
a	0000
b	0001
c	0010
d	0011
e	0100
f	0110
g	1000

We insert the records in the order given above. Show the extensible hash index after all records have been inserted.

Solution: As is the case with extensible hash indices, you do not have to consider the sequence by which the records are inserted. The resulting index is



3 Algebra (10 points)

Consider the following two definitions of the semijoin operator $\rhd<$, which may or may not be equivalent.

- **Direct** (from page 255 of the textbook) The semijoin $\ltimes$ of relations R and S, written $R \ltimes S$, is the bag of tuples t in R such that there is at least one tuple in S that agrees with t in all attributes that R and S have in common.
- **Indirect** Let us call $a(R)$ the list of attributes of R. Then, it is

  $$R \ltimes S = \pi_{a(R)}(R \bowtie S)$$

  where $\bowtie$ stands for the natural join.

1. **5 points** Are the two definition equivalent? Assume bag semantics for the algebra. If the answer is yes provide proof, showing that, given arbitrary R and S, if a tuple t appears k times in the result of $R \ltimes S$ according to the direct definition, then the tuple t will appear k times in the result of $R \ltimes S$ according to the indirect definition. If the answer is no, provide an example with an R table and an S table and show $R \ltimes S$ for the direct and the indirect definition.

   Solution: No, they are not equivalent. Consider the following counterexample with relations $R(A)$ and $S(A, B)$:

   $$R = \begin{array}{|c|}\hline A \\ \hline 1 \\ \hline \end{array}$$

   and

   $$S = \begin{array}{|c|c|}\hline A & B \\ \hline 1 & 2 \\ \hline 1 & 3 \\ \hline \end{array}$$

   Then accoding to the direct definition

   $$R \ltimes S = \begin{array}{|c|}\hline A \\ \hline 1 \\ \hline \end{array}$$

   But according to the indirect definition

   $$R \ltimes S = \pi_A \left(\begin{array}{|c|c|}\hline A & B \\ \hline 1 & 2 \\ \hline 1 & 3 \\ \hline \end{array} \right) = \begin{array}{|c|}\hline A \\ \hline 1 \\ \hline 1 \\ \hline \end{array}$$

2. **5 points** Consider the indirect definition of $\ltimes$. Assume that the schema of P is $P(A, B, C, D)$ and the schema of T is $T(C, E)$. Prove the following, using the notation shown in the Appendix.

   $$\pi_{A,B}\sigma_{D=5 \wedge E=6}(P \bowtie T) = \pi_{A,B}[(\pi_{A,B,C}\sigma_{D=5}P) \ltimes (\pi_C \sigma_{E=6}T)]$$

Solution: Exercising transformation rules from the notes and the book we have

$$\begin{aligned}
&\pi_{A,B}\sigma_{D=5\wedge E=6}(P \bowtie T) = \\
&\pi_{A,B}\sigma_{D=5}\sigma_{E=6}(P \bowtie T) = \\
&\pi_{A,B}\sigma_{D=5}(P \bowtie \sigma_{E=6}T) = \\
&\pi_{A,B}(\sigma_{D=5}(P) \bowtie \sigma_{E=6}(T)) = \\
&\pi_{A,B}(\pi_{A,B,C}\sigma_{D=5}(P) \bowtie \pi_C\sigma_{E=6}(T)) = \\
&\pi_{A,B}\pi_{A,B,C}(\pi_{A,B,C}\sigma_{D=5}(P) \bowtie \pi_C\sigma_{E=6}(T)) = \\
&\pi_{A,B}((\pi_{A,B,C}\sigma_{D=5}P) \ltimes (\pi_C\sigma_{E=6}T))
\end{aligned}$$

4 Query Processing (32 points)

Consider the relations $Cust(CID, Name, City)$, $Order(OID, CID, Date)$, and $LineItem(LID, OID, Product, Amount)$, where CID is a customer id and is a key for $Cust$, OID is an order id and is a key for $Order$, and LID is a line item id and is a key for $LineItem$. In addition the attribute CID of $Order$ is a foreign key referring to the CID of $Cust$, that is, for each CID c of $Order$ there is exactly one tuple of $Cust$ whose CID attribute is c. The OID of $LineItem$ is a foreign key referring to the OID of $Order$.

Assume the following statistics, where all numbers, except for *product*, correspond to millions.

$T(Cust) = 1$	$V(Cust, CID) = 1$
	$V(Cust, Name) = 0.5$
$T(Order) = 20$	$V(Order, OID) = 20$
	$V(Order, CID) = 1$
$T(LineItem) = 100$	$V(LineItem, LID) = 100$
	$V(LineItem, OID) = 20$
	$V(LineItem, Product) = 1000$

Consider the following SQL query, which returns the total amount for each product that "Jones" bought.

```
SELECT Product, SUM(Amount) AS Total
FROM Cust, Order, LineItem
WHERE Cust.CID = Order.CID AND Order.OID = LineItem.OID AND Cust.Name = 'Jones'
GROUPBY Product
```

1. **5 points** Write an algebra expression that uses two cartesian products $\times$, exactly one selection σ and the $SUM_{Product;Amount\mapsto Total}$ operator and computes the SQL query.

   Solution: Let us denote "Customer" by C, "Order" by O, and "LineItem" by L. Then it is

$$SUM_{Product;Amount\mapsto Total}\sigma_{C.CID=O.CID\wedge O.OID=L.OID\wedge C.Name='Jones'}((C \times O) \times L)$$

2. **5 points** Show the series of transformations that transform the algebra expression of the previous question into an expression where the cartesian products have been replaced by joins and the selections are pushed as down (early) as possible.

   Assume that the equation $\sigma_{R.A=S.A}(R \times S) = R \bowtie_{R.A=S.A} S$ is one of the transformation rules.

   Solution:

   $$\begin{aligned}
   &SUM_{Product;Amount\mapsto Total}\sigma_{C.CID=O.CID \wedge O.OID=L.OID \wedge C.Name='Jones'}((C \times O) \times L) = \\
   &SUM_{Product;Amount\mapsto Total}\sigma_{C.CID=O.CID}\sigma_{O.OID=L.OID \wedge C.Name='Jones'}((C \times O) \times L) = \\
   &SUM_{Product;Amount\mapsto Total}\sigma_{C.CID=O.CID}\sigma_{O.OID=L.OID}\sigma_{C.Name='Jones'}((C \times O) \times L) = \\
   &SUM_{Product;Amount\mapsto Total}\sigma_{C.CID=O.CID}\sigma_{O.OID=L.OID}(\sigma_{C.Name='Jones'}(C \times O) \times L) = \\
   &SUM_{Product;Amount\mapsto Total}\sigma_{C.CID=O.CID}\sigma_{O.OID=L.OID}((\sigma_{C.Name='Jones'}(C) \times O) \times L) = \\
   &SUM_{Product;Amount\mapsto Total}\sigma_{C.CID=O.CID}((\sigma_{C.Name='Jones'}(C) \times O) \bowtie_{O.OID=L.OID} L) = \\
   &SUM_{Product;Amount\mapsto Total}(\sigma_{C.CID=O.CID}(\sigma_{C.Name='Jones'}(C) \times O) \bowtie_{O.OID=L.OID} L) = \\
   &SUM_{Product;Amount\mapsto Total}((\sigma_{C.Name='Jones'}(C) \bowtie_{C.CID=O.CID} O) \bowtie_{O.OID=L.OID} L) =
   \end{aligned}$$

3. **5 points** Provide an additional (non-trivial) expression where the selections have been pushed as down (early) as possible, but the join order is different. No need to show the series of transformations that led you to this expression.

   Solution:

   $$SUM_{Product;Amount\mapsto Total}(\sigma_{C.Name='Jones'}(C) \bowtie_{C.CID=O.CID} (O \bowtie_{O.OID=L.OID} L)) =$$

4. **10 points** So far you must have provided 2 algebra expressions with different join orders. For each one of them provide an estimate of the size of its intermediate results. Also, provide an estimate of the size of the final result. To save time, just put the size numbers next to the edges in the algebra expressions.

   Solution: Let's start with the first plan

   $$\begin{aligned}
   &T(\sigma_{C.Name='Jones'}C) = \frac{1M}{0.5M} = 2 \\
   &T(\sigma_{C.Name='Jones'}(C) \bowtie_{C.CID=O.CID} O) = 2\frac{20M}{1M} = 40 \\
   &T((\sigma_{C.Name='Jones'}(C) \bowtie_{C.CID=O.CID} O) \bowtie_{O.OID=L.OID} L) = 40\frac{100M}{20M} = 200
   \end{aligned}$$

   And now the second plan

   $$\begin{aligned}
   &T(\sigma_{C.Name='Jones'}C) = \frac{1M}{0.5M} = 2 \\
   &T(O \bowtie_{O.OID=L.OID} L) = T(L) = 100M \\
   &V(O \bowtie_{O.OID=L.OID} L, O.CID) = 1M \\
   &T((\sigma_{C.Name='Jones'}C) \bowtie_{C.CID=O.CID} (O \bowtie_{O.OID=L.OID} L)) = 2\frac{100M}{1M} = 200
   \end{aligned}$$

   If you reached that point you have got full points. So, what about the $SUM_{Product;Amount\mapsto Total}$? The book provides upper and lower bound numbers as well as some pretty arbitrary estimates for the duplicate elimination operation, which is identical to the problem at hand. If you are a combinatorics freak you will recognize Stirling's numbers as the solution. More in class!

5. **7 points** Assume that you have got indices on all attributes. Consider the following simplification of the query:

```
SELECT Product, Amount
FROM Cust, Order, LineItem
WHERE Cust.CID = Order.CID AND Order.OID = LineItem.OID AND Cust.Name = 'Jones'
```

Apply the INGRES algorithm to get a join order. Indicate which is the small relation hyperedge that you pick in each step of the algorithm and show the resulting plan.

A What will get you full points in proving equivalence of algebraic expressions

Assume that the exercise is to prove that

$$\sigma_{p \wedge q}(P \bowtie T) = (\sigma_p P) \bowtie (\sigma_q T)$$

where p refers to attributes of P only and q refers to attributes of T only.

The cleanest proofs are the ones where each step corresponds to application of one of the rules in the notes. In our example, it goes as follows:

$$\begin{array}{l} \sigma_{p \wedge q}(P \bowtie T) = \\ \sigma_p \sigma_q (P \bowtie T) = \\ \sigma_p (P \bowtie \sigma_q T) = \\ (\sigma_p P) \bowtie (\sigma_q T) \end{array}$$