Assignment 4: Indexing and Query Processing (Spring 2019)

Instructor: Zhaonian Zou (`znzou@hit.edu.cn`)

Name: ____________ Student ID: ____________ Grade: ______

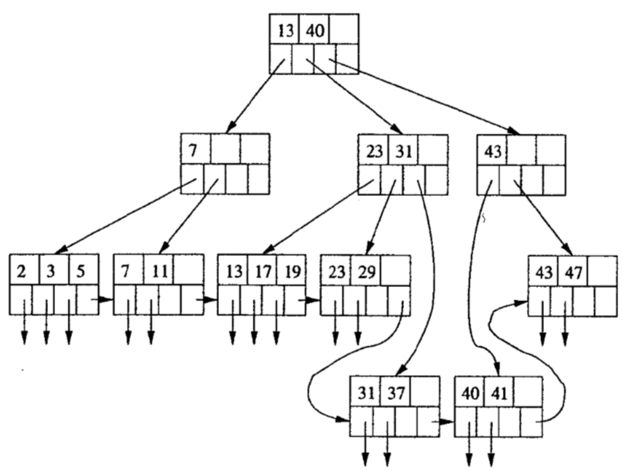
Question	1	2	3	4	5	6	7	Total
Score								

Notes

- **Print the assignment on A4 paper and answer the questions.**
- **Assignment due date: May 18/19, 2019 (the 2nd lab).**

Questions

1. (20 Points) We have a B+-tree depicted as follows.



Complete the following operations on the tree.

   (a) (10 Points) Insert a tuple with search key 12. Describe the insertion process and draw the B+-tree obtained after the insertion.

   (b) (10 Points) Delete the tuple with search key 40. Describe the deletion process and draw the B+-tree obtained after the deletion.

2. (10 Points) Design a one-pass algorithm to implement the group-by operation $\gamma_{A;sum(B)}(R)$ and analyze the I/O cost and memory requirement of the algorithm.

3. (10 Points) Design a hash-based algorithm to implement the group-by operation $\gamma_{A;sum(B)}(R)$ and analyze the I/O cost and memory requirement of the algorithm.

4. (10 Points) Design a sort-based algorithm to implement the group-by operation $\gamma_{A;sum(B)}(R)$ and analyze the I/O cost and memory requirement of the algorithm.

5. (10 Points) Suppose there is a covering index on attributes A and B for relation R. Design an algorithm to implement the group-by operation $\gamma_{A;sum(B)}(R)$, which utilizes the covering index. Analyze the I/O cost and memory requirement of the algorithm.

6. (20 Points) We are given the statistics of 4 relations $W(a, b)$, $X(b, c)$, $Y(c, d)$, and $Z(d, e)$.

$W(a,b)$	$X(b,c)$	$Y(c,d)$	$Z(d,e)$
$T(W) = 100$	$T(X) = 200$	$T(Y) = 300$	$T(Z) = 400$
$V(W, a) = 20$	$V(X, b) = 50$	$V(Y, c) = 50$	$V(Z, d) = 40$
$V(W, b) = 60$	$V(X, c) = 100$	$V(Y, d) = 50$	$V(Z, e) = 100$

   (a) (5 Points) Estimate the cost of the following relational-algebra expression

$$\Pi_{b,c,d,e}(\sigma_{a=10 \wedge e>0}(W \bowtie X \bowtie Y \bowtie Z)).$$

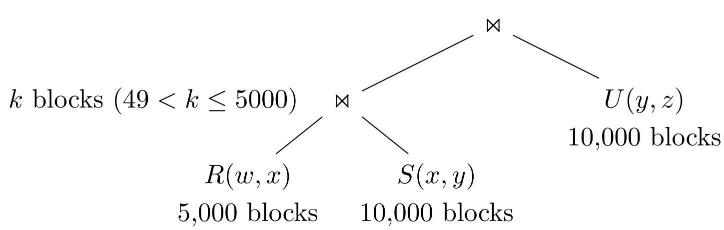
   (b) (5 Points) Transform the expression to an equivalent one that has lower estimated cost and give the estimated cost.

   (c) (10 Points) Determine the best order for evaluating $W \bowtie X \bowtie Y$, using left-deep join trees only.

7. (20 Points) Consider the following relational-algebra experession. The input relations $R(w, x)$, $S(x, y)$, and $U(y, z)$ are stored on disk in 5,000, 10,000, and 10,000 blocks, respectively. We are going to execute this expression using the following execution plan:

   - The operation $R \bowtie S$ is executed using the hash-join algorithm.
   - The join operation on $(R \bowtie S)$ and U is executed using the nested-loop join algorithm.
   - The tuples in $R \bowtie S$ are piplined to the join operation on $(R \bowtie S)$ and U.

   Suppose there are $M = 101$ blocks in the buffer pool available for executing the experession, and the tuples in $R \bowtie S$ occupy k blocks, where $49 < k \leq 5000$.



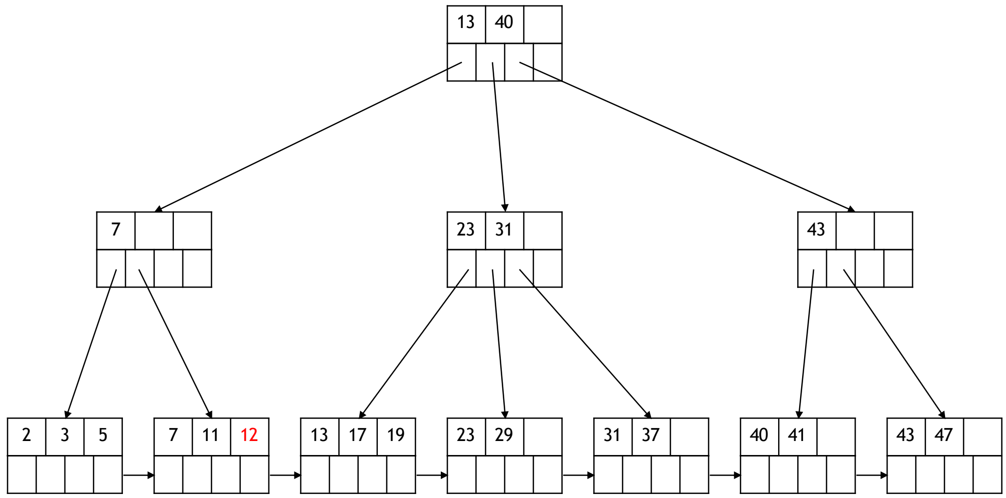
   Answer the following questions.

   (a) (10 Points) How many times has relation U been scanned during the execution of $(R \bowtie S) \bowtie U$?

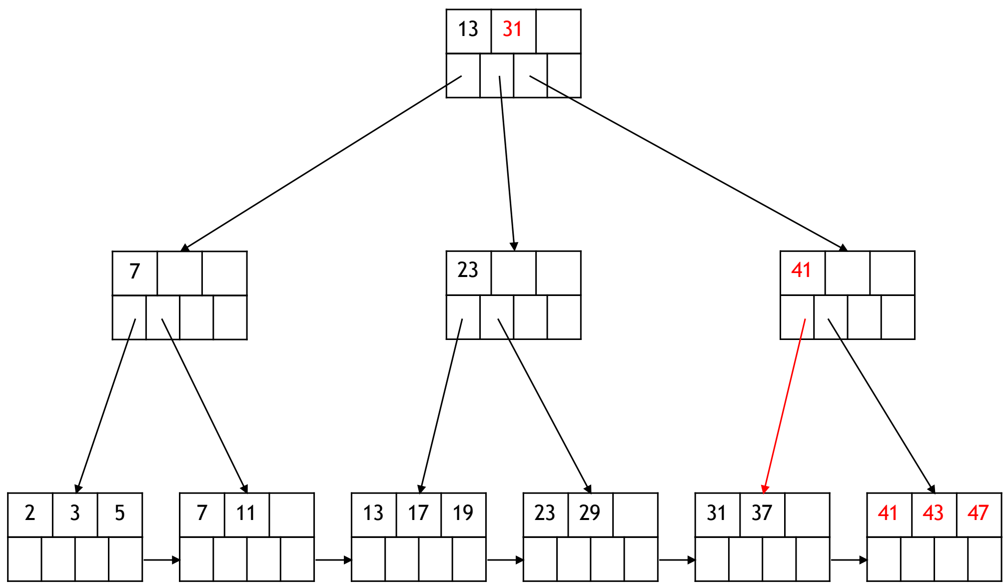
   (b) (10 Points) Analyze the I/O cost for executing the expression according to the given plan.

Answers

1. (a) After inserting search key 12, the B+-tree is depicted as follows.



   (b) After deleting search key 40, the B+-tree is depicted as follows.



2. The one-pass aggregation algorithm is very similar to the *one-pass duplicate elimination algorithm*. The details of the algorithm is omitted. The I/O cost of the algorithm is $B(R)$. The number, M, of buffers available for use in the buffer pool must satisfy $M \geq B(R) + 1$.

3. The hash-based aggregation algorithm is very similar to the *hash-based duplicate elimination algorithm*. The details of the algorithm is omitted. The I/O cost of the algorithm is $3B(R)$. The number, M, of buffers available for use in the buffer pool must satisfy $B(R) \leq (M-1)^2$.

4. The sort-based aggregation algorithm is very similar to the *sort-based duplicate elimination algorithm*. The details of the algorithm is omitted. The I/O cost of the algorithm is $3B(R)$. The number, M, of buffers available for use in the buffer pool must satisfy $M \geq \sqrt{B(R)}$.

5. The I/O cost is the number of blocks for storing the index. The memory requirement is $M \geq 1$.

6. (a) • $T(W \bowtie X) = T(W)T(X)/\max(V(W,b), V(X,b)) = 20000/60$.

- $T((W \bowtie X) \bowtie Y) = T(W \bowtie X)T(Y)/\max(V(X,c), V(Y,c)) = 1000$.
- $T(((W \bowtie X) \bowtie Y) \bowtie Z) = T((W \bowtie X) \bowtie Y)T(Z)/\max(V(Y,d), V(Z,d)) = 8000$.
- $T(\sigma_{a=10 \wedge e>0}(W \bowtie X \bowtie Y \bowtie Z)) = T(W \bowtie X \bowtie Y \bowtie Z) \cdot \frac{1}{V(W,a)} \cdot \frac{1}{3} = 8000/60$.
- $\Pi_{b,c,d,e}(\sigma_{a=10 \wedge e>0}(W \bowtie X \bowtie Y \bowtie Z)) = T(\sigma_{a=10 \wedge e>0}(W \bowtie X \bowtie Y \bowtie Z))/2 = 4000/60$.

The estimated cost is the total size of intermediate results, that is, $20000/60 + 1000 + 8000 + 8000/60$.

(b) $\Pi_{b,c,d,e}(\sigma_{a=10}(W) \bowtie X \bowtie Y \bowtie \sigma_{e>0}(Z))$. The cost for this plan can be estimated similarly.

(c)
- The estimated cost for evaluating $(W \bowtie X) \bowtie Y$ is

$$T(W \bowtie X) = T(W)T(X)/\max(V(W,b), V(X,b)) = 20000/60.$$

- The estimated cost for evaluating $(W \bowtie Y) \bowtie X$ is

$$T(W \bowtie Y) = T(W)T(Y) = 30000.$$

- The estimated cost for evaluating $(X \bowtie Y) \bowtie W$ is

$$T(X \bowtie Y) = T(X)T(Y)/\max(V(X,c), V(Y,c)) = 600.$$

Therefore, the best order for evaluating $W \bowtie X \bowtie Y$ is $(X \bowtie Y) \bowtie W$.

7. (a) When evaluating $R_i \bowtie S_i$ using the one-pass join algorithm, 49 blocks are used for storing R_i, and 1 block is used for inputting S_i. Hence, there are 50 blocks available for piplining $R \bowtie S$ and evaluating $R \bowtie S \bowtie U$. Among these 50 blocks, 49 blocks are used for piplining the tuples in $R \bowtie S$, and 1 block is used for inputting U. Therefore, relation U is scanned $\lceil k/49 \rceil$ times.

(b)
- The I/O cost for hashing R is 5000.
- The I/O cost for hashing S is 10000.
- The total I/O cost for joining all pairs of R_i and S_i is 15000.
- The tuples in $R \bowtie S$ are piplined, so no I/Os.
- Relation U is scanned $k/49$ times, so the I/O cost for scanning U is $\lceil 10000k/49 \rceil$.

Thus, the I/O cost for this plan is $30000 + 10000\lceil k/49 \rceil$.