CSI 410. Database Systems – Spring 2021

Homework Assignment I

The deadline for this assignment is **11:59 PM, February 26, 2021**. *Submissions after this deadline will not be accepted.* Each student is required to enter the UAlbany Blackboard system and then upload a .pdf file (in the form of [first name]_[last name].pdf) that contains answers to Problems 1-3.

The total grade for this assignment is 100 points. If you find any error or have questions or suggestions, please contact the instructor (`jhh@cs.albany.edu`).

Problem 1. (20 points) Consider a database buffer manager that uses the algorithm explained in Section 13.5.1 of the textbook for a hard drive which takes 10 milliseconds per disk seek and can read 80MB per second.

(a) (10 points) Assume that (i) the size of each disk block is 8KB, (ii) the probability that the buffer manager can handle a data request without accessing the disk (i.e., the probability that a requested block is already in the buffer) is 90%, (iii) only 10% of disk block accesses require a disk seek (i.e., 90% of the disk blocks are read consecutively without requring any disk seek), (iv) no disk blocks are updated (i.e., no need to write dirty blocks back to the hard drive), and (v) the time spent for reading/writing data in the main memory is negligible (considered 0 milliseconds). Under these assumptions, calculate the expected data access time (i.e., how much time it would on average take to obtain a disk block from the buffer manager).

(*Answer*) With 90% probability, the requested disk block is in the buffer (no disk access is needed). With 9% probability (i.e., 90% of 10%), the requested disk block is not in the buffer and can be read on disk without any disk seek, incurring $\frac{8KB}{80MB/sec} = 0.1$ milliseconds. With 1% probability (i.e., 10% of 10%), the requested disk block is not in the buffer and reading that block on disk requires a disk seek. The time cost in this case is 10 milliseconds + $\frac{8KB}{80MB/sec}$ = 10.1 milliseconds. Therefore, the overall expected data access time is 0.11 milliseconds ($= 0.09 \cdot 0.1 + 0.01 \cdot 10.1$ milliseconds).

(b) (10 points) In contrast to the above scenario, assume that (i) the size of each disk block is 16KB, (ii) the probability that the buffer manager can handle a data request without accessing the disk (i.e., the probability that a requested block is already in the buffer) is 70%, and (iii) only 5% of disk block accesses require a disk seek (i.e., 95% of the disk blocks are read consecutively without requring any disk seek). Calculate the expected data access time as in (a). Also, based on the above calculations, explain whether it is more advantageous to use 8KB disk blocks or 16KB disk blocks.

(*Answer*) With 70% probability, the requested disk block is in the buffer (no disk access is needed). With 28.5% probability (i.e., 95% of 30%), the requested disk block is not in the buffer and can be read on disk without any disk seek, incurring $\frac{16KB}{80MB/sec} = 0.2$ milliseconds. With 1.5% probability (i.e., 5% of 30%), the requested disk block is not in the buffer and reading that block on disk requires a disk seek. The time cost in this case is 10 milliseconds + $\frac{16KB}{80MB/sec}$ = 10.2 milliseconds. Therefore, the overall expected data access time is 0.21 milliseconds ($= 0.285 \cdot 0.2 + 0.015 \cdot 10.2$ milliseconds).

When only a small portion of each disk block (e.g., a record) is in general needed, it is more advantageous to use 8KB disk blocks. If the entirety of each disk block is needed, it is more beneficial to use 16KB disk blocks (equivalent to 0.105 milliseconds per 8KB).

Problem 2. (40 points) Consider the following relational database:

```
branch(branch_name, branch_city)
customer(customer_number, customer_name, customer_city)
loan(loan_number, branch_name, amount)
borrower(customer_number, loan_number)
```

(a) (10 points) Identify an appropriate primary key for each of the above relations. Assume that (i) each branch is assigned a unique name, (ii) each customer is assigned a unique number, (iii) each loan is assigned a unique number, (iv) a customer may have multiple loans and a loan may be shared by multiple customers.

(*Answer*) The primary key for the `branch` relation is {`branch_name`} since each branch is assigned a unique name. Similarly, the primary key for the `customer` relation is {`customer_number`} and the primary key for the `loan` relation is {`loan_number`}. The primary key for the `borrower` relation is {`customer_number`, `loan_number`} since each combination of a customer number and a loan number is unique. Note that `customer_number` must be included in the primary key because, as the table below shows, there can be multiple tuples (the second and third tuples) with the same `loan_number` (L2). Similarly, `loan_number` must be included in the primary key because there can be multiple tuples (the first and second tuples) with the same `customer_number` (C1).

customer_number	loan_number
C1	L1
C1	L2
C2	L2

(b) (10 points) Using the schema of the `customer` relation, provide an example of a superkey which is not a candidate key. Explain why your answer is correct.

(*Answer*) In the `customer` relation, {`customer_number`, `customer_name`} is a superkey, but is not a candidate key because {`customer_number`} is also a superkey.

(c) (10 points) Given your choice of primary keys, identify all of the foreign keys. For each foreign key, specify the referencing and referenced relations.

(*Answer*)

foreign key	referencing relation	referenced relation
{branch_name}	loan	branch
{customer_number}	borrower	customer
{loan_number}	borrower	loan

(d) (10 points) For one of the foreign keys identified above, explain a situation where deleting a record/tuple causes a violation of the foreign key constraint (referential integrity constraint).

(*Answer*) In the `loan` relation, foreign key {`branch_name`} references the `branch` relation. The foreign key constraint requires that every value of the `branch_name` attribute in the `loan` relation also appear in the `branch` relation. Suppose that the `branch` relation contains a record (B1, Albany) and the `loan` relation contains a record (L1, B1, 10000). Deleting (B1, Albany) from `branch` violates the foreign key constraint mentioned above.

Problem 3. (40 points) Answer the following problems. **For each problem, start with the B+-tree in Figure 1**.

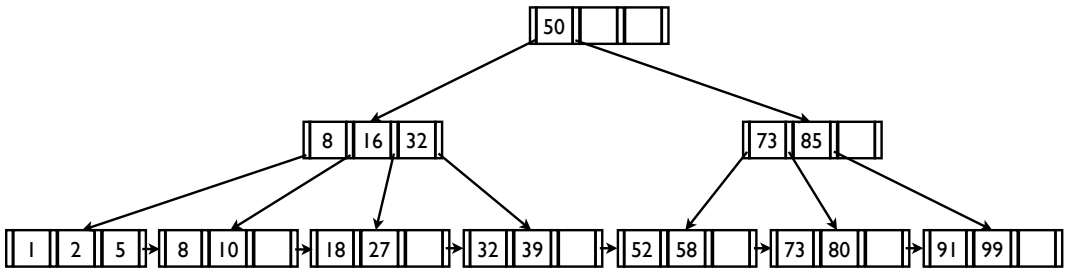
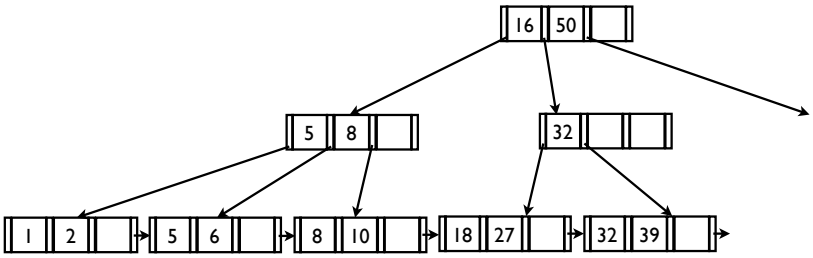


Figure 1: B+-Tree Example

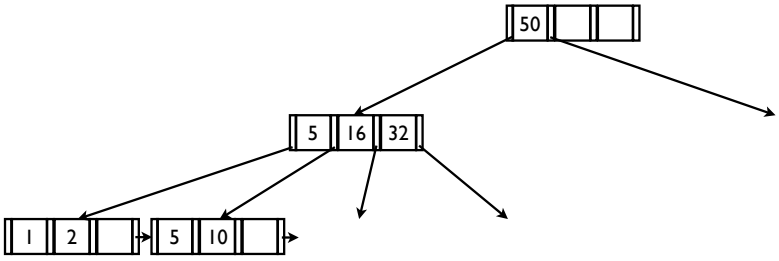
(a) (10 points) Draw the tree that would result from inserting 6 into the tree in Figure 1. When a node is split into two nodes, ensure that the right node has no more keys/pointers than the left node. You may omit the parts of the tree that do not change.

(*Answer*)



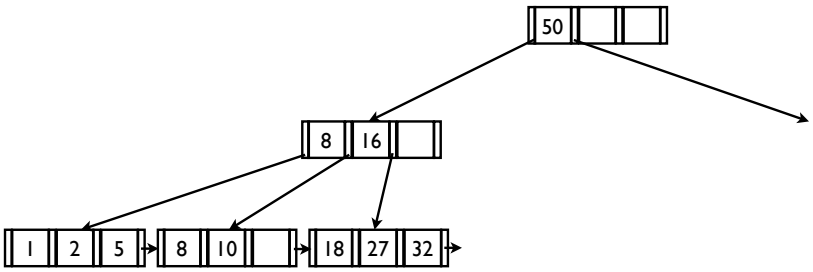
(b) (10 points) Draw the tree that would result from deleting 8 from the tree in Figure 1. When node merging or redistribution is needed, if both the left and right sibling nodes are available, use the left sibling node. You may omit the parts of the tree that do not change.

(*Answer*)



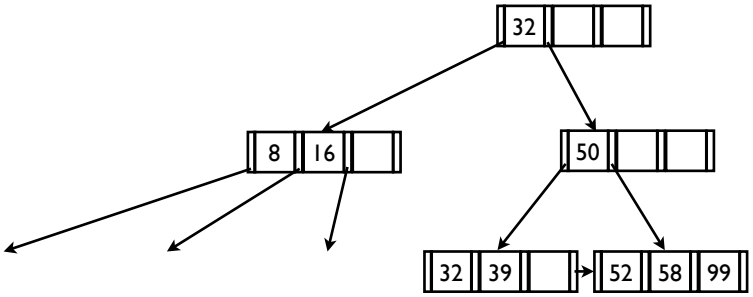
(c) (10 points) Draw the tree that would result from deleting 39 from the tree in Figure 1. You may omit the parts of the tree that do not change.

(*Answer*)



(d) (10 points) Draw the tree that would result from deleting 91, 80, and 73 from the tree in Figure 1. You may omit the parts of the tree that do not change.

(*Answer*)



It should be noted that every leaf node must have at least two key values and every non-leaf node must have at least two children.

After solving the above problems, please state the amount of time spent for this assignment. Feel free to add comments or suggestions if any.