Name

CWID

Exam

May 11th, 2021, 8:00 - 10:00am

CS525 - Spring 2021

Please leave this empty! 1.1 □ 1.2 □ 2 □ 3 □ Sum □

Instructions

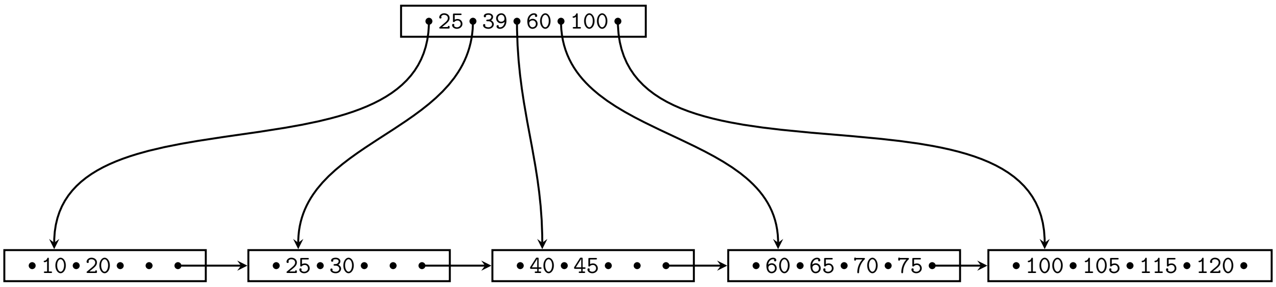
- **You have to hand in the exam via course blackboard**
- **This is an individual and not a group assignment. Fraud will result in 0 points**

BY SUBMITTING THIS EXAM THROUGH THE ONLINE SYSTEM, I AFFIRM ON MY HONOR THAT I AM AWARE OF THE STUDENT DISCIPLINARY CODE, AND (I) HAVE NOT GIVEN NOR RECEIVED ANY UNAUTHORIZED AID TO/FROM ANY PERSON OR PERSONS, AND (II) HAVE NOT USED ANY UNAUTHORIZED MATERIALS IN COMPLETING MY ANSWERS TO THIS TAKE-HOME EXAMINATION.

Part 1 Index Structures (Total: 55 Points)

Question 1.1 B$^+$ Index Structures (30 Points)

Consider the following `B+-tree` ($n = 4$).



Draw the modified tree after **insert 80**, **delete 60**, **delete 25**, and then **insert 110**.
You need to show the resulted tree after performing each operation.

When splitting or merging nodes follow these conventions:

- **Leaf Split**: In case a leaf node needs to be split, the left node should get the extra key if the keys cannot be split evenly.
- **Non-Leaf Split**: In case a non-leaf node is split evenly, the "middle" value should be taken from the right node.
- **Node Underflow**: In case of a node underflow you should first try to redistribute and only if this fails merge. Both approaches should prefer the left sibling.

Scratch Page - This page is intentionally left blank

Scratch Page - This page is intentionally left blank

Question 1.2 Linear Hash Table (25 Points)

Make the following assumptions:

- A bucket can hold five index entries.
- We start with a hash table with two empty blocks (corresponding to `0` and `1`)
- We insert records with keys `0000,0001,0010,..., 1111`
- Assume that a linear hash table is used to index the database with a capacity threshold of **85%**.
- Show the index structure after the insertions.
- In order to get full credit, you MUST show your work

Scratch Page - This page is intentionally left blank

Scratch Page - This page is intentionally left blank

Part 2 Result Size Estimations (Total: 20 Points)

Consider a table `book` with attributes ISBN, title, author, edition (primary key is ISBN), a table `library` with loc, budget, public (primary key is loc), and a table `catalog` with attributes library and book.
`catalog`.library is a foreign key to relation `library`.
Attribute book of relation `catalog` is a foreign key to relation `book`.
Given are the following statistics:

T(book)=	100,000	T(library)=	100	T(catalog)=	200,000
V(book, ISBN)=	100,000	V(library, loc)=	100	V(catalog, library)=	100
V(book, title)=	50,000	V(library, budget)=	40	V(catalog, book)=	90,000
V(book, author)=	30,000	V(library, public)=	2		
V(book, edition)=	10				

Question 2.1 Estimate Result Size (4 Points)

Estimate the number of result tuples for the query $q = \sigma_{public=true \wedge budget \leq 10}(\texttt{library})$ using the first assumption presented in class (values used in queries are uniformly distributed within the active domain).

Question 2.2 Estimate Result Size (5 Points)

Estimate the number of result tuples for the query $q = \sigma_{title='Faust' \wedge author='Goethe' \wedge edition='3'}(\texttt{book})$ using the first assumption presented in class. Assume that the minimal and maximal values in the `edition` attribute are `1` and `10`.

Question 2.3 Estimate Result Size (5 Points)

Estimate the number of result tuples for the query $q = \sigma_{edition \geq 3 \wedge edition \leq 8 \wedge title='Databases'}(\texttt{book})$ using the first assumption presented in class. Assume that the minimal and maximal values in the `edition` attribute are 1 and 10.

Question 2.4 Estimate Result Size (6 Points)

Estimate the number of result tuples for the query
$q = \sigma_{title='Int.\ to\ DB' \vee title='Database\ Systems'}(\texttt{book}) \bowtie_{\texttt{title=book}} \texttt{catalog} \bowtie_{\texttt{library=loc}} \sigma_{\texttt{budget} \leq \texttt{40}}(\texttt{library})$
using the first assumption presented in class.

Part 3 I/O Cost Estimation (Total: 25 Points)

Question 3.1 I/O Cost Estimation (25 Points)

Consider two relations R and S with B(R)=2,000,000 and B(S)=200,000 blocks, and S(R)=$\frac{1}{400}$, and S(S)=$\frac{1}{200}$. You have M = 200 memory pages available. Let R has an index on the joining attribute C. Compute the minimum number of I/O operations needed to join these two relations using:

1. **tuple-based-nested-loop join**: Relation R is clustered, relation S is not clustered.
2. **block-nested-loop join**: Relations R and S are clustered.
3. **merge-join**: The inputs are not sorted. Relation R is clustered, relation S is not clustered.
4. **clustering index-join**: Relations are clustered but not sorted. Read of S with uniform distribution assumption and expected 200 matching tuples on R for S.
5. **hash-join**: Relations are clustered but not sorted. You can assume that the hash function evenly distributes keys across buckets.

Justify you result by showing the I/O cost estimation for each join method.

Scratch Page - This page is intentionally left blank

Scratch Page - This page is intentionally left blank