Name	CWID

Quiz
2

November 21-30, 2018

CS525 - Quiz 2
Solutions

Please leave this empty!	1		2		3		Sum	

Instructions

- **Due on Blackboard by November 30, 2018 11:59PM Chicago Time**
- This is an individual and not a group assignment. Fraud will result in **0** points
- Things that you are allowed to use
  - Textbook
  - Lecture notes (electronic or printed)
  - Personal notes
- For your convenience the number of points for each part and questions are shown in parenthesis.
- There are **3** parts in this exam
  1. Relational Algebra
  2. Result Size / I-O Cost Estimation
  3. Index Structures
- `I affirm my awareness of the standards of the Illinois Institute of Technology Honor Code`

`Sign ..............................................................`

Part 1 Relational Algebra (Total: 15 Points)

Question 1.1 (15 Points)

Consider relations $R(A, B)$, $S(B, C, D)$ and $T(C, D)$. The following sub-problems ask you to rewrite relational algebra expressions. You can assume that the relations contain sets (not bags). If the requested rewrite is not feasible, state so and briefly explain why. Also, make sure there are no unneeded expressions in your rewrite, e.g., $\pi_{CD}(T)$ and $\sigma_A \neq A(R)$ are unneeded.

(a) State whether the following expression is feasible. If so, rewrite the following expression (including the projection, if necessary) by pushing the projection as far down as possible:

$\pi_{AD}[\sigma_{C=5}(R \bowtie S)]$

Solution

$\pi_{AD}[R \bowtie \pi_{BD}(\sigma_{C=5}S)]$

(b) State whether the following expression is feasible. If so, rewrite the following expression so it does not contain a union operator and contains one selection operator (instead of two):

$[R \bowtie (\sigma_{C=2}S)] \cup [(\sigma_{A=1}R) \bowtie S]$

Solution

$\sigma_{(C=2 \vee A=1)}[R \bowtie S]$

(c) State whether the following expression is feasible. If so, What is the minimum number of operators required to express the query represented by the following expression?

$\sigma_{D=2}[(\sigma_{C=2}S) \bowtie (\sigma_{C=1}T)]$

Solution

Not feasible or can say zero; if we select all rows with $C = 2$ from S and all rows with $C = 1$ from T, then join on C and D via natural join, we are guaranteed to have no output so we can avoid running the query entirely.

Part 2 Result Size Estimations (Total: 20 Points)

Consider two relations `R(A, B, C)` and `S(B, C, D)`. We want to estimate the number of tuples and the size of the following expression: $\mathbf{U} = \pi_{ACD}\left[(\sigma_{A=3 \wedge B=5}\mathbf{R}) \bowtie \mathbf{S}\right]$

We are given the following information:

- T(R) = 100000; V(R, A) = 20; V(R, B) = 50; V(R, C) = 150.
- T(S) = 5000; V(S, B) = 100; V(S, C) = 200; V(S, D) = 30.
- All attributes are 10 bytes in size.
- We assume query values are selected from values in the relations.

Question 2.1 Estimate Result Size (6 Points)

First consider the innermost select $\mathbf{W} = \sigma_{A=3 \wedge B=5}\mathbf{R}$. Compute the following values.

Solution

1. T(W) $= T(R) \times \frac{1}{V(R,A)} \times \frac{1}{V(R,B)} = 100000 \times \frac{1}{20} \times \frac{1}{50} = 100$

2. S(W) $=$ `Size of tuple of a relation` W `= # of attributes` $\times$ `attribute size` $=3 \times 10 = 30$ bytes

3. V(W, A) = `1 (Attribute A in the relation W has only the value 3)`

4. V(W, B) `= 1 (Attribute B in the relation W has only the value 5)`

5. V(W, C) = `100` $\big($`The maximum possible value for V(W,C) is T(W)`$\big)$

Question 2.2 Estimate Result Size (8 Points)

Next consider the join $\mathbf{Y} = \mathbf{W} \bowtie \mathbf{S}$. Compute the following values

Solution

1. T(Y) $= \frac{T(W)\times T(S)}{\max\big(V(W,B),V(S,B)\big)\max\big(V(W,C),V(S,C)\big)} = \frac{T(W)\times T(S)}{V(S,B)\times V(S,C)} = \frac{100\times 5000}{100\times 200} = 25$

2. S(Y) = `Size of tuple of a relation` Y `= # of attributes` × `attribute size` $=4\times 10 = 40$ bytes

3. V(Y, A) = `1` (`Attribute A in the relation Y has only the value 3`)

4. V(Y, B) `= 1` (`Attribute B in the relation Y has only the value 5`)

5. V(Y, C) = `25` (`The maximum possible value for V(Y,C) is T(Y)`)

6. V(Y, D) = `25`

Question 2.3 Estimate Result Size (6 Points)

Finally consider the full expression $\mathbf{U} = \pi_{ACD}\left[(\sigma_{A=3 \wedge B=5}\mathbf{R}) \bowtie \mathbf{S}\right]$. Compute the following values.

1. T(U) = 25

2. S(U) = 30

3. V(U, A) = 1

4. V(U, C) = 25

5. V(U, D)= 25

Question 2.4 I/O Cost Estimation (15 Points)

Assume a database system that holds two important relations, `R` and `S`, that are frequently joined over a common attribute `A`. The relations are currently stored as rows.

- Relation `R` has three attributes, `A`, `B`, and `C` where each `10` bytes long.
- Relation `S` has three attributes, `A`, `D`, and `E`, where `A` is 10 bytes long and `D` and `E` each `15` bytes long.
- Each of R and S contain `64000` tuples.
- In addition to its header, each disk block can hold `6400` bytes.
- To perform `R` $\bowtie$ `S` we use a simple hash join algorithm as described in our class notes. Assume there is enough main memory.
- The expected number of resulting tuples in `R` $\bowtie$ `S` is `10`.
- The tuples of relation R are stored contiguously in blocks. They are not sorted.
- The S tuples are also contiguous, spanned, and unsorted.

What is the number of IOs needed for the hash join? (Do not include the cost of writing the final result to disk.)

Solution

Each tuple is 30 bytes since each attribute is 10 bytes each, and there are 3 attributes in each tuple.
There are 42000 tuples in each relation which means there are 42000 × 30 bytes per relation.
Since each block can hold 4200 bytes, $B(R) = B(S) = \frac{42000 \times 30}{4200} = 300$ blocks.
`Hash join` requires each relation to be read in, written out into hash buckets, and then read in one final time for the join phase. Thus, `total cost`$=3 \times \big(B(R) + B(S)\big) = 3 \times (300 + 300) = 3 \times 600 = 1800$ Disk IOs.

Part 3 Index Structures (Total: 20 Points)

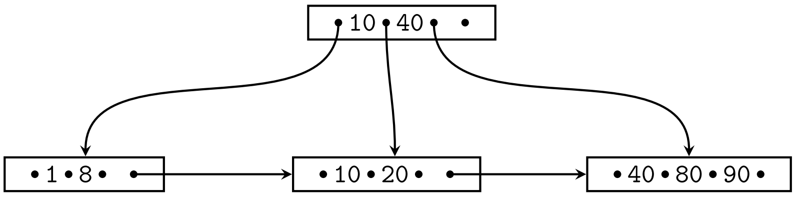
Question 3.1 B+-tree Operations (20 Points)

Given is the B+-tree shown below ($n = 3$). Execute the following operations and write down the resulting B+-tree after each step:

insert(15), insert(30), insert(110), delete(30), delete(10), delete(80)

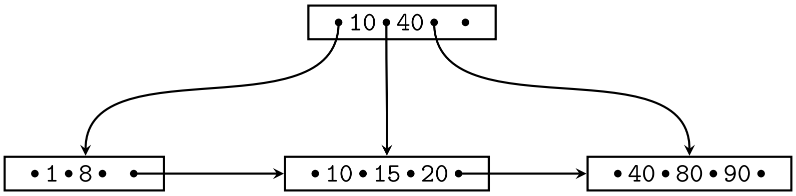
When splitting or merging nodes follow these conventions:

- **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.

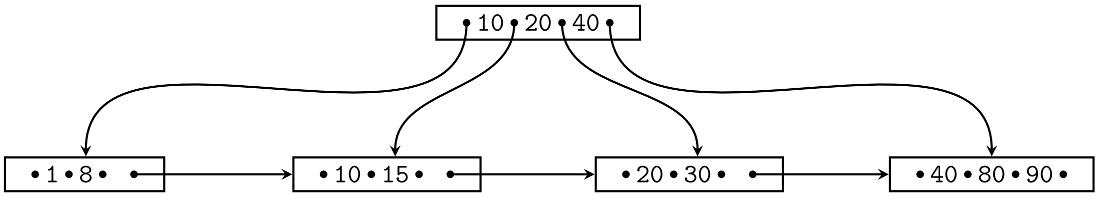


Solution

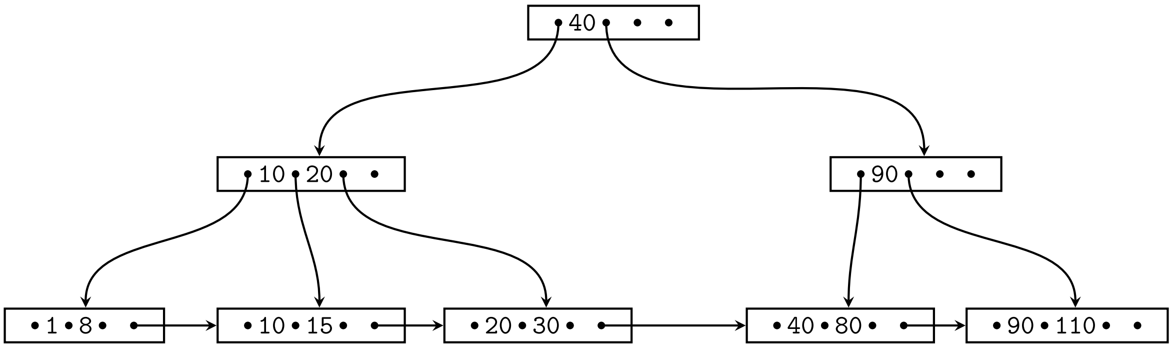
insert(15)



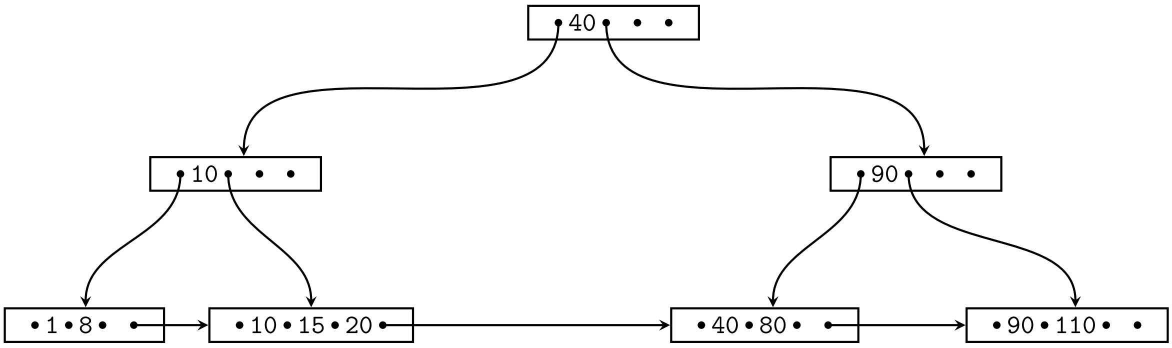
insert(30)



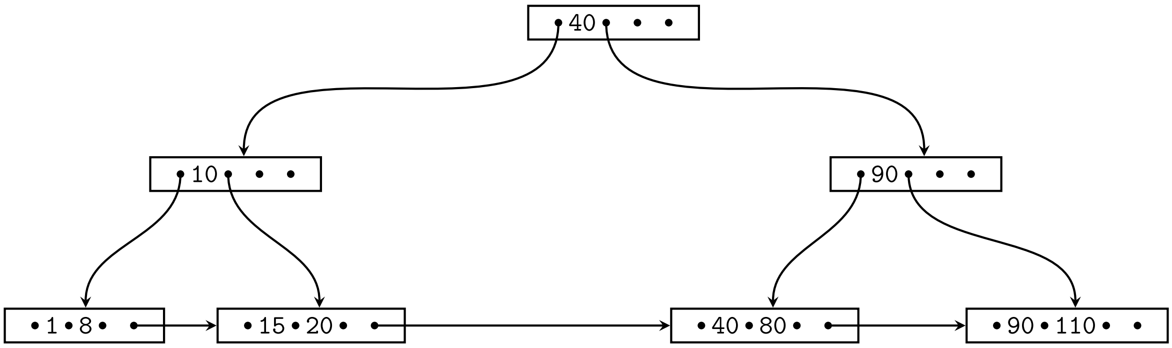
insert(110)



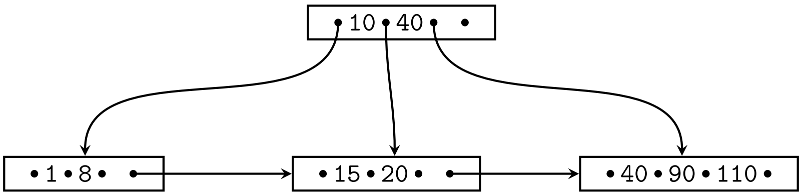
delete(30)



delete(10)



delete(80)



Question 3.2 Extensible Hash Operations (10 Points)

Consider an `extensible hash` structure with the following characteristics:

- Buckets can hold up to two records.
- No overflow blocks are allowed.
- The hash function we use generates `b = 4` bits total.
- Initially the extensible hash table is empty.

Say we insert `X` records, where the search key of each record generates a distinct 4-bit hash value (no collisions). No records are deleted during this process. We are told that after the `X` insertions, 4 buckets have been allocated. (Note that the previous sentence does not refer to the size of the directory.)

4.1. What is the minimum possible value of `X`?

```
Minimum value of X: 3
```

4.2. In the same scenario, what is the maximum possible value of `X`?

```
Maximum value of X: 8
```