CS 1555 – Database Management Systems (Fall 2020)

Dept. of Computer Science, University of Pittsburgh

Practice Assignment: Data Storage, Indexing & Concurrency Control

Release: Nov. 21, 2020

Goal

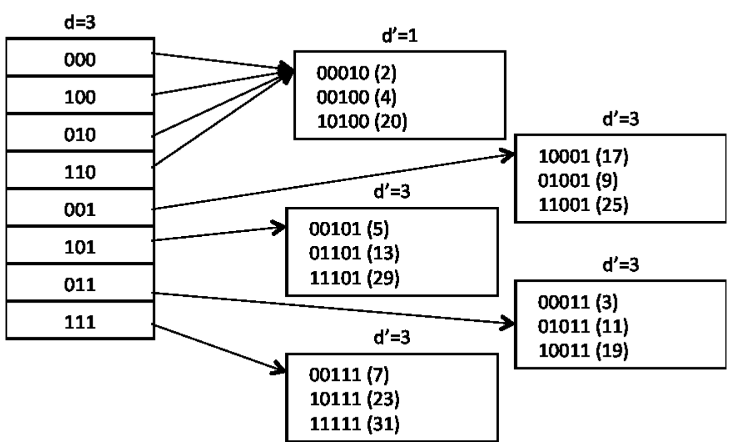
The goal of this assignment is to understand and gain familiarity with data storage of database records, index structures and to review the concepts of concurrency control **as part of the preparation for the final exam**.

Questions

1. [20 points] Show the extensible hashing structure that results from inserting the below keys using the least significant bits. Suppose that buckets can hold three records.

   2, 3, 5, 7, 11, 17, 9, 13, 4, 19, 20, 29, 31, 25, 23

   Answer:

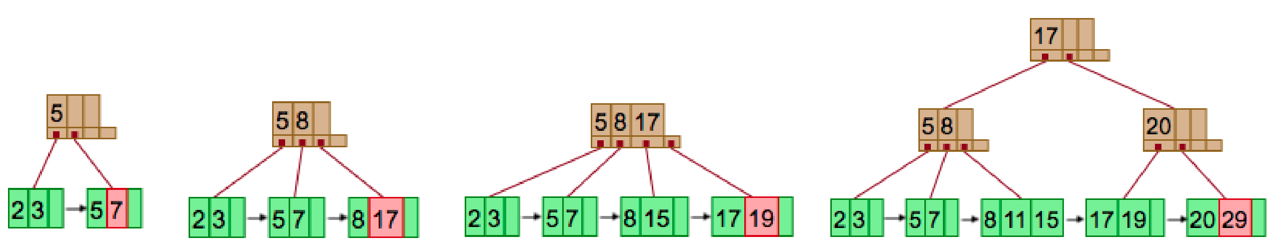


2. Using the *class method* (not the textbook method), construct a B$^+$-tree for the following set of key values:

   2, 3, 5, 7, 8, 17, 15, 19, 11, 20, 29.

   Assume that the tree is initially empty and values are added in the above order. Assume that the order of the tree is 4, that is, the number of pointers that will fit in one node is four. Show the B$^+$-tree after inserting 7, 17, 19 and 29.

   Answer:



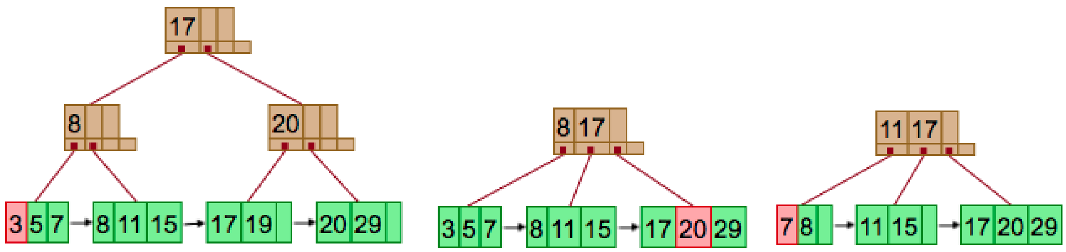
3. Answer the following questions:

   (a) Show how the B$^+$-tree you created in Question 1 changes when you delete key 2.

   (b) Show how the B$^+$-tree changes when you further delete key 19.

   (c) Show how the B$^+$-tree changes when you further delete keys 3 and 5.

   Answer:



4. Consider a table $R(A, B)$ with 5127 rows, where B is an alternative key, i.e., a unique key but not the primary key with no NULL values. Suppose that each B$^+$-tree index block can hold up to 8 keys and 9 pointers. What is the minimum number of levels required for a B$^+$-tree index on attribute B? Explain how you got your answer. The root counts as a level.

   Answer:

   Note that the minimum number is the optimal case in which all leaf nodes are fully packed.
   The minimum number of levels required for a B+-tree index on attribute B is 4 levels.

   Every block can hold up to 8 keys and points, so every block can have up to 9 children. The first level (root) can have up to 8 keys. The second level can have up to 9 blocks and a total of 9*8 keys. The third level can have up to 9*9 blocks and a total of 9*9*8 keys (= 648 keys). The fourth level can have up to 9*9*9 blocks and a total of 9*9*9*8 keys (= 5832 keys). Since we have 5127 keys, the minimum number of levels required for a B+-tree index on attribute B is 4.

5. Concurrency Control - Serializability

   Consider the following Histories:

$$H_1 = R_3(z)W_2(x)R_2(z)W_2(y)R_1(x)R_3(x)R_3(y)W_1(x)$$

$$H_2 = W_2(x)W_2(y)W_1(x)R_3(x)R_1(x)R_3(z)R_3(y)R_2(z)$$

   (a) List the conflicts in each history.
   Answer: We consider pairs of operations from left to right.
   $H_1 : R_3(z)W_2(x)R_2(z)W_2(y)R_1(x)R_3(x)R_3(y)W_1(x)$

   - $R_3(z)$ does not conflict with any other operation
   - $W_2(x) \rightarrow R_1(x); W_2(x) \rightarrow R_3(x); W_2(x) \rightarrow W_1(x);$
   - $R_2(z)$ does not conflict with any other operation
   - $W_2(y) \rightarrow R_3(y)$
   - $R_1(x)$ does not conflict with any other operation
   - $R_3(x) \rightarrow W_1(x)$
   - $R_3(y)$ does not conflict with any other operation
   - $W_1(x)$ does not conflict with any other operation

$H_2 = W_2(x)W_2(y)W_1(x)R_3(x)R_1(x)R_3(z)R_3(y)R_2(z)$

- $W_2(x) \rightarrow W_1(x); W_2(x) \rightarrow R_3(x); W_2(x) \rightarrow R_1(x);$
- $W_2(y) \rightarrow R_3(y)$
- $W_1(x) \rightarrow R_3(x)$
- $R_3(x)$ does not conflict with any other operation
- $R_1(x)$ does not conflict with any other operation
- $R_3(z)$ does not conflict with any other operation
- $R_3(y)$ does not conflict with any other operation
- $R_2(z)$ does not conflict with any other operation

(b) H_1 is serializable. Which of the following serial histories HS_1=T_2, T_1, T_3, and HS_2=T_2, T_3, T_1 is equivalent to H_1

Answer: Find the conflicts in each of these serial histories and compared them with those of H_1.

$\text{HS}_1 = T_2, T_1, T_3 = W_2(x)R_2(z)W_2(y)R_1(x)W_1(x)R_3(z)R_3(x)R_3(y)$

- $W_2(x) \rightarrow R_1(x); W_2(x) \rightarrow R_3(x); W_2(x) \rightarrow W_1(x);$
- $R_2(z)$ does not conflict with any other operation
- $W_2(y) \rightarrow R_3(y)$
- $R_1(x)$ does not conflict with any other operation
- $W_1(x) \rightarrow R_3(x)$
- $R_3(z)$ does not conflict with any other operation
- $R_3(x)$ does not conflict with any other operation
- $R_3(y)$ does not conflict with any other operation

H_1 does not have the same conflicts with HS_1 and hence they are not equivalent.

HS_2=$T_2, T_3, T_1 = W_2(x)R_2(z)W_2(y)R_3(z)R_3(x)R_3(y)R_1(x)W_1(x)$

- $W_2(x) \rightarrow R_1(x); W_2(x) \rightarrow R_3(x); W_2(x) \rightarrow W_1(x);$
- $R_2(z)$ does not conflict with any other operation
- $W_2(y) \rightarrow R_3(y)$
- $R_3(z)$ does not conflict with any other operation
- $R_3(x) \rightarrow W_1(x)$
- $R_3(y)$ does not conflict with any other operation
- $R_1(x)$ does not conflict with any other operation
- $W_1(x)$ does not conflict with any other operation

H_1 has the same conflicts with HS_2 and hence they are equivalent.

(c) Which of them could have been generated by 2PL schedule.

Answer: H_1 could have been generated by 2PL schedule, because once the scheduler has released a lock for a transaction, it does not request any additional locks on any data item for that transaction.