EECS 495
Introduction to Database Systems
Fall 2018
Instructor: Mas-ud Hussain
Homework Assignment No. 4
Due: Sunday, December 09, 2018

Problem No. 1:

Consider an extendible hashing index and a linear hashing index, that are initialized with four buckets (2 bits) each and have a hash bucket size of four entries. For each of the following problems, insert a set of data entries into both indexes such that the given property is satisfied. Draw the state of the indexes after the entries have been inserted. Note that you must insert the same set of data entries into both indexes.

(a) The extendible hashing index has more pages than the linear hashing index.
(b) For both indexes, inserting any data entry will increase the number of pages.
(c) Both indexes have an equal number of pages.

Problem No. 2:

Consider a B+ tree index. The keys are integer values; the leaf node of the index stores keys and pointer to the actual data location. The index implements the following variant of deletion. (This is full deletion as we discussed in class.) If a node N becomes less than half full, N first checks whether re-distribution can take place with its right sibling R. If re-distribution with R is not possible, then N checks whether it can merge with R. If neither re-distribution nor merge is possible with node R, or if node N does not have a right sibling, then N tries the same with its left sibling, in the same order first check whether re-distribution is possible, then check whether merge is possible.) The height of a B+ tree is the length of the longest path from the root to any leaf node. As an example, a B+ tree that contains only of the root has height one, a B+ tree that consists of a root node and two leaf nodes has height 2.

Suppose, the B+ tree for this problem uses the rule of "*Minimum 50% occupancy (except for root)*". That is, each non-leaf and leaf node (except root) contains **n/2 <= *m* <= n** keys (a.k.a. data entries). Note that, this rule is slightly different from the rules in the lecture file, where we had different "min. values" rules for leaf and non-leaf nodes. For this problem, the value of **n is 4** – i.e., each non-leaf and leaf node except root must contain between 2 and 4 keys. If the number of keys become >4, a split is necessary. If the number of keys become <2, a redistribution or merge is necessary. The root must have at least 1 key. Given these settings and rules, answer the following questions.

(a) Assume that the index has the following two properties: (i) if you insert a specific additional data entry (i.e., key) into the index, the height of the index would increase from 2 to 3, (ii) if you delete that same data entry from the index, the height of the index will not change. Draw a B+ tree index that has these two properties, give the data entry that you would insert, and explain why the properties hold.
(b) Draw the B+ tree index from above after the insertion and deletion of your new data entry. Assume that a split leaf results in a 3-2 distribution of entries over the new and old leaf nodes.
(c) Assume that you want to decrease the height of your new B+ tree index back to 2. List the elements you would have to delete and draw the B+ tree index after those elements have been deleted.

Problem No. 3:

Assume, we are joining the following tables:

A=8,20,19,20,3,13,20,18,6,5,4,5
B=15,1,3,13,13,10,19,6,8,15,16,2

If the in-memory runs are of length, m = 4, and we are applying the sort-merge algorithm to perform the join:

(a) what will the sorted runs be for A and B?
(b) What is the join output? Walk through the steps of the merge.

Problem No. 4:

A data warehouse of a telephone provider consists of five dimensions: *caller customer, callee customer, time, call type, and call program*, and three measures: *number of calls, duration, and amount*. Both the caller and callee customer dimensions store customer street address information with city, state, country information. The time dimension itself stores information in day, month, quarter, and year granularity.

(a) Create the MultiDim representation for the conceptual schema of the warehouse (show appropriate dimension hierarchies as described above).
(b) Based on the multi-dim representation above, assume that the cube is named "Calls" and the cube initially starts at the bottom level (i.e., leaf) of each possible dimension hierarchy. Define the OLAP operations to be performed to answer the following queries:
    1. Total duration of calls made by customers from Brussels in 2016.
    2. Total amount collected by each call program in 2016.
    3. Total number of weekend calls made by customers from Brussels to customers in Antwerp in 2016.
    4. Total duration of international calls started by customers in Belgium in 2016.