Braden Ackles

CS 440

1/21/2017

Homework #2

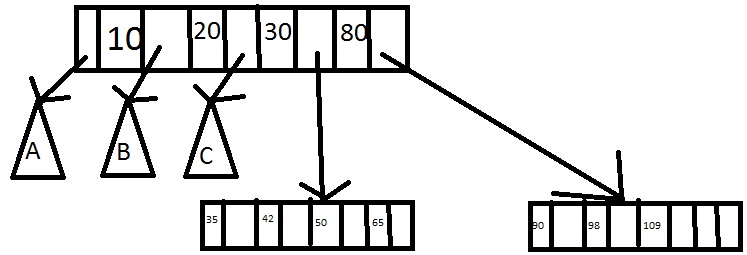
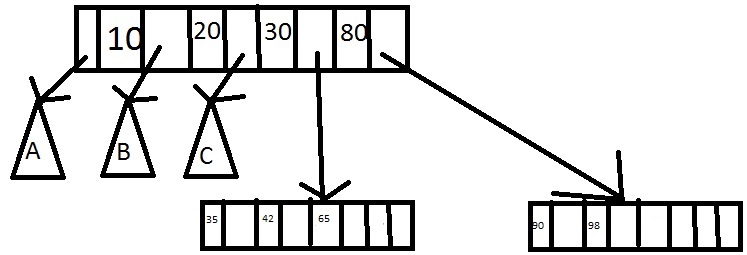
Problem #1:

Consider a file with a large number of Person(id, name, birth-date) records. Assume that users frequently search this file based on a the field id to find the values of name or birth- date for people whose information is stored in the file. Moreover, assume that users rarely update current records or insert new records to the file. Which file structure, heap versus sorted, provides the fastest total running time for users’ queries over this file? Explain your answer (1 point).

**Answer #1:** The fastest file structure is sorted file structure**.** This is because in the sorted file system the data will be stored logically making it easier to find and reducing the number of operations needed to find the correct value. A heap would not be faster because it is stored in memory blocks and it would be almost random and has the ability to be faster if it is one of the first values in the heap but most likely will take much longer than the sorted file structure. A sorted file structure also is very efficient and fast and finding the correct results bye to its nature being ordered and there for a logical comparing processes able to narrow down the sorting required.

Problem #2:

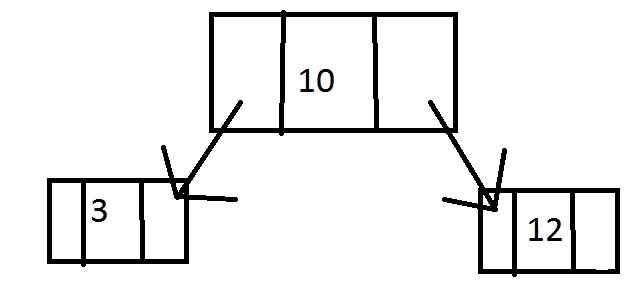
Consider the B+ tree index shown in Figure 1. Each intermediate node can hold up to five pointers and four key values. Each leaf can hold up to four pointers to data, and leaf nodes are doubly linked as usual, although these links are not shown in the figure. Note that A, B and C are subtrees of the B+ tree, but are not fully specified. Answer the following questions (2 points).

1. Show the B+ tree that would result from inserting a record with a search key 109 into the tree.
   1. 
2. Show the B+ tree that would result from deleting the record with search key 50 from the original tree
   1. 
3. Name a search key value such that inserting it into the (original) tree would cause an increase in the height of the tree.
   1. Any values between 65 and 79 would cause a change in height because the list will already have five elements and there for need to change heights.
4. What can you infer about the contents and shape of A,B and C subtrees?
   1. That their values fall between their respective places. For example in A all values are less than ten. In B all values are greater than ten and less than 20. For C all values are between 20 and 30.

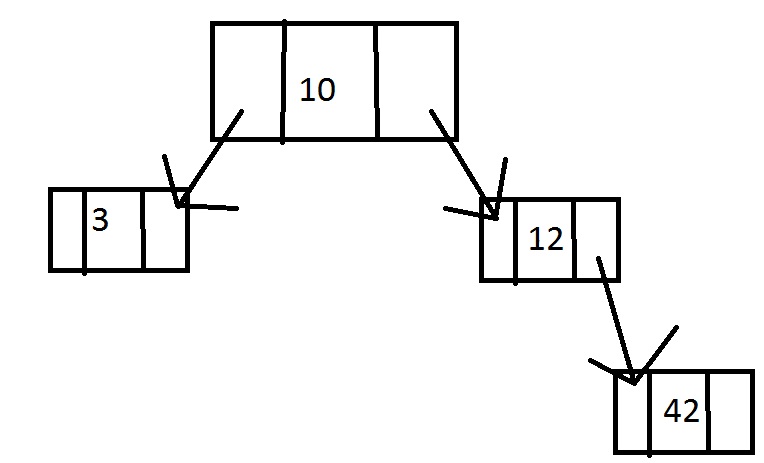
Problem #3:

Suppose that a block can contain at most four data values and that data values are integers. Using only B+ trees of degree 2, give examples of each of the following (1.5 points)

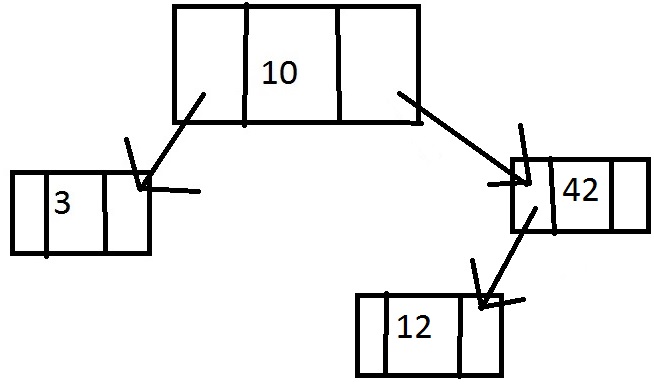
1. A B+ tree whose height changes from 2 to 3 when the value 42 is inserted. Show your structure before and after the insertion
   1. Before



After

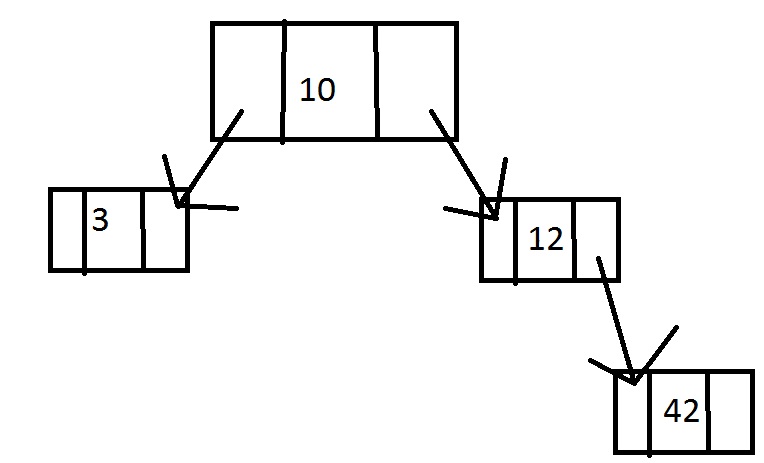


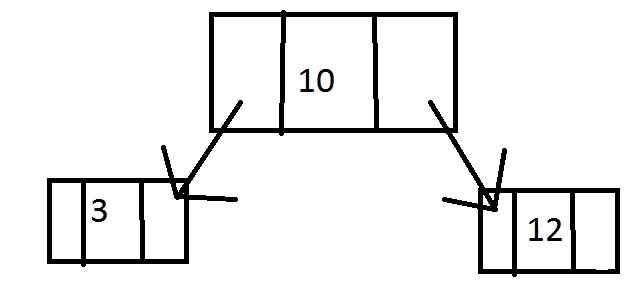
1. A B+ tree in which the election of value 42 leads to a redistribution. Show your structure before and after the deletion.
   1. Before:



After:

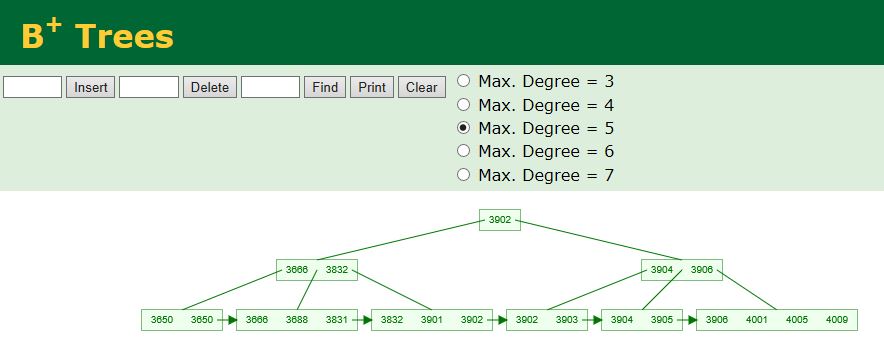
Since 42 is not a leaf switch positions with its leaf and then delete





Problem #4:

Consider the instance of the Students relation shown in Figure 2 and address the following questions (1.5 points).

1. To reduce the number of I/O access in index search, each B+ tree node fit in a block. Let Sid be an integer requiring 32 bits. Let a pointer require 64 bits. If the block size is 56 bytes (son siting of 8 bits), what should be the minimum degree of the B+ tree index on Sid?
   1. The minimum degree for the B+ tree should be 3. This is the most efficient tree possible. With the fewest I/O operations.
2. Show a B+ tree on Sid of degree calculated in part 1 for all records in Figure 2
   1. 

I found a tool for doing B+ trees when I was doing problem #4 however it only lets you do 4 digit numbers I know all numbers in the table have a preceeding 5 so I just wanted to put that here.