UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Assignment 4 - Report

Department of Industrial and Enterprise Systems Engineering
Zhenye Na (zna2)
April 17, 2018

1 INDEXING (10 PTS)

1. Suppose we want to build an index on a relation R which has a total of x records, with each block capable of holding either y records or z key-pointer pairs. Assuming x is divisible by y, please answer the following questions (if your value evaluates to a fraction, use ceiling $\lfloor\ \rfloor$ or floor $\lceil\ \rceil$ as appropriate):

   a) Suppose you construct a simple single level index, and that index is dense. How many index blocks are required to access all of the records of R?
   Solutions:
   We need $\lceil \frac{x}{z} \rceil$ index blocks.

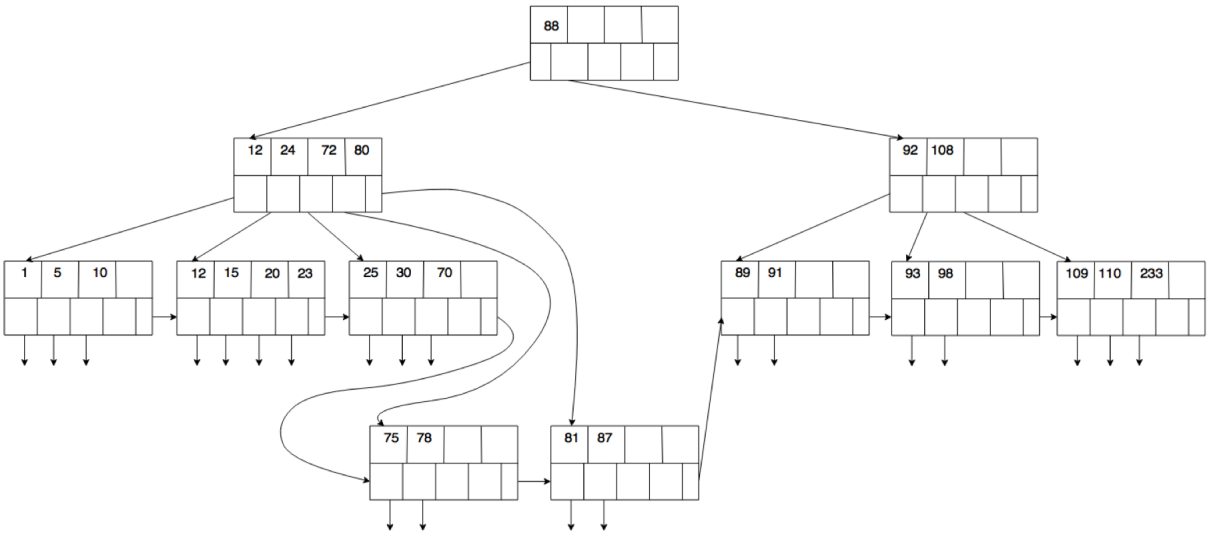
   b) Suppose the index built is sparse. If the index stores a pointer to the lowest search key in each block, and the index is a simple single level index, how many data blocks do we need? How many index blocks do we need?
   Solutions:
   We need $\left\lceil \frac{x}{yz} \right\rceil$ index blocks and $\left\lceil \frac{x}{y} \right\rceil$ data blocks.

2. True/False question - In order to use a dense index, you will have to have the data file sorted by the search key; otherwise, you will need to use a sparse index. Explain your reasoning.
   Solutions: False. First, in order to use dense index, you do not *have to* sort the index. You can directly use an unclustered index, which you *have to* use dense index.

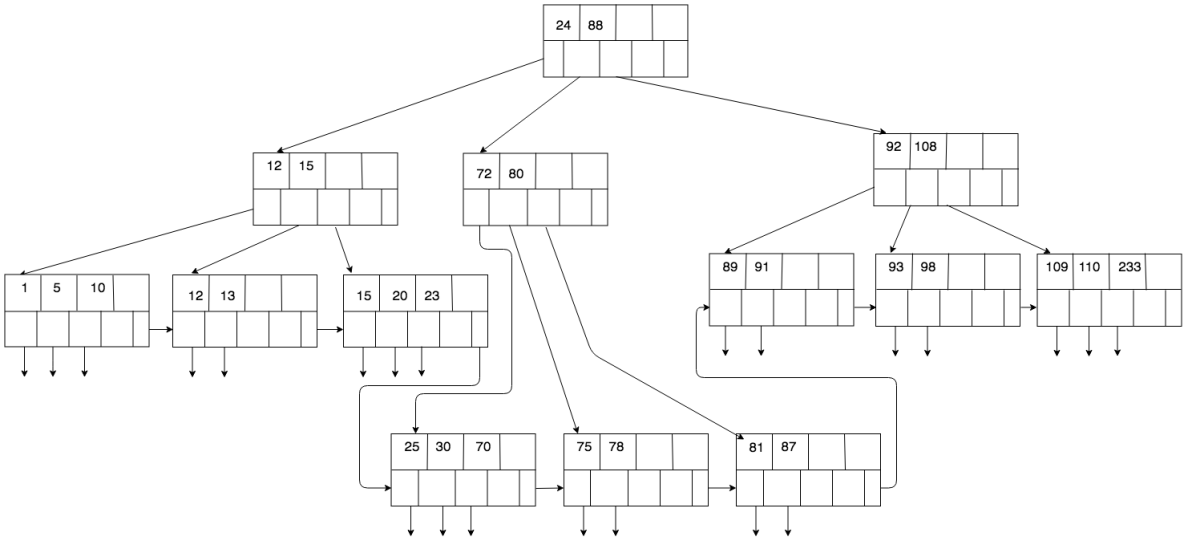
2 B+ TREE (30 PTS)

In this question, I kept the first two questions as the previous version and only changed the third question to the new version of question.

Consider a B+ tree of degree 2 shown below:

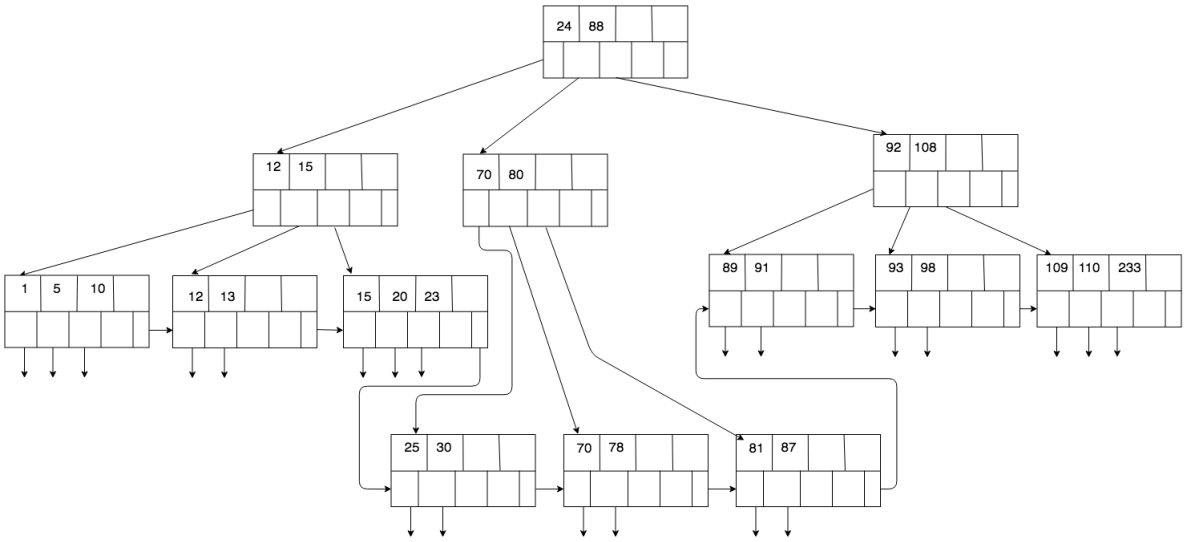


1. Draw the B+ tree that would result from inserting a data entry with key 13.
   Solutions:

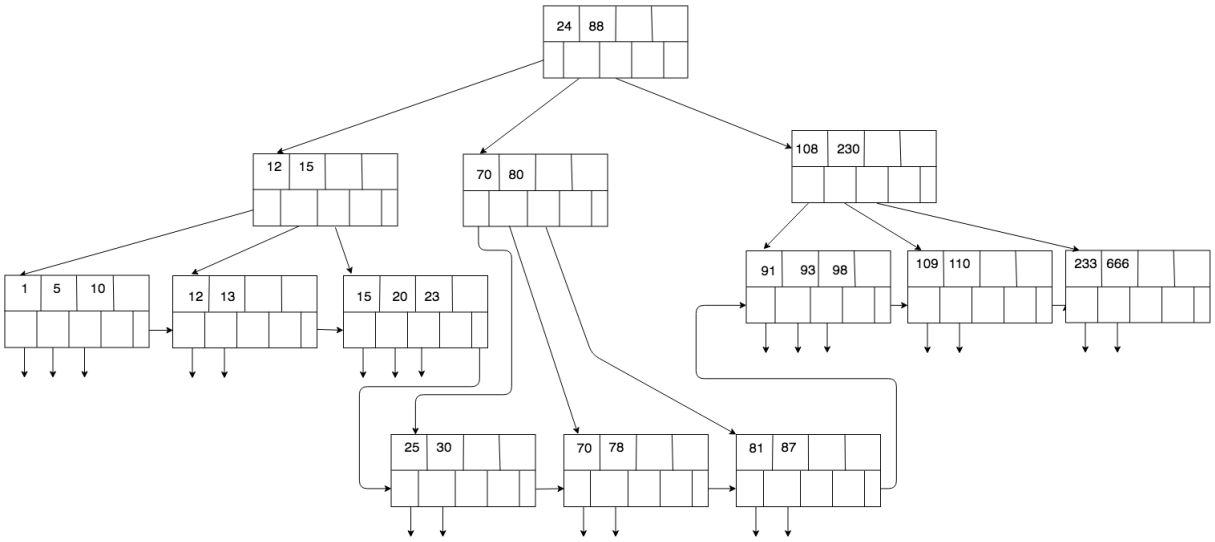


2. Based on the B+ tree that you drew in the previous question, draw the B+ tree that would result from deleting the data entry with key 75.

Solutions:



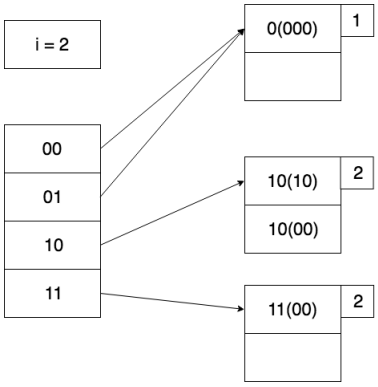
3. Based on the B+ tree that you drew in the previous question, draw the B+ tree that would result from deleting the data entry with key 89.
Solutions:



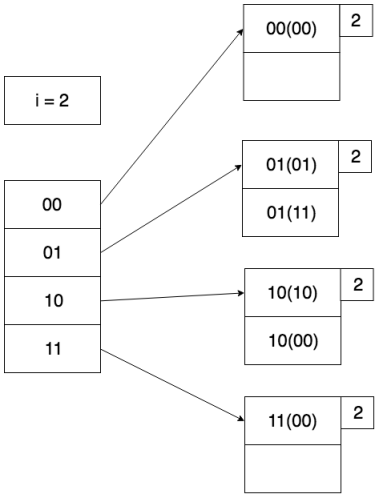
3 Extensible Hashing (30 pts)

Assume you have a extensible hash table with hash function h(k) = k mod 13, expressed as a binary string of size 4, and data block of size 2 (i.e., it can accommodate two tuples). You are asked to index the following key values in order: 25, 13, 23, 21.

1. Draw the extensible hash table which obeys the above constraints after the four keys are inserted.
   Solutions: First, we apply hash function to every key value.
   25 -> 12 -> 1100
   13 -> 0 -> 0000
   23 -> 10 -> 1010
   21 -> 8 -> 1000



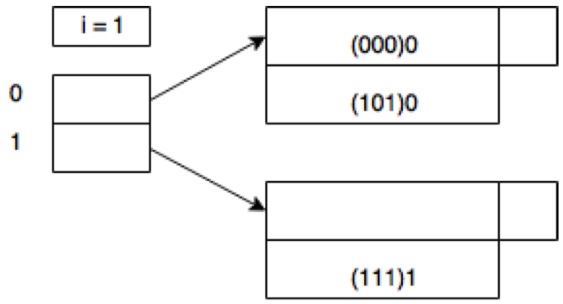
2. Using your solution to the previous question, now consider insertion of keys 18 and 20 into the hash table, and draw the resulting hash table.
   Solutions: First, we apply hash function to every key value.
   18 -> 5 -> 0101
   20 -> 7 -> 0111



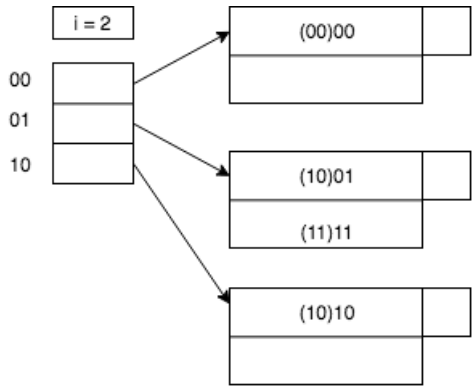
4 LINEAR HASHING (30 PTS)

Consider a linear hash table with $r \leq 1.76n$ with each data block capable of holding 2 records (that is, the average number of record per bucket should not exceed 88% of the total number

of records per block):



1. Insert 1001 and draw the resulting table.
   Solutions:



2. With your solution from the previous question, insert 1101, 1110, 0001 incrementally and draw the final table; that is, insert one at a time, check the condition, and move to the next one.

Solutions: