CS411: Database Systems **Spring 2017**

Assignment 4 Solutions

Question 1. Indexing (10 points)

Assume our DBMS is implemented on a system with block size of 32kb. We want to build an index on a relation R with a total of 1,000,000 records, where each record is 256 bytes and each key/pointer pair requires 8 bytes.

1. Suppose the index is dense and unclustered. How many blocks are required to store all the records of R (data file and index file)?
   Answer: *assuming* 32kb = 32,000 bytes
   We need a (key, pointer) pair for each record. The total number of index blocks + number of data blocks

$$(1,000,000/(32kB/8B)) + (1,000,000/(32kB/256B)) = 8250 \text{ Blocks}$$

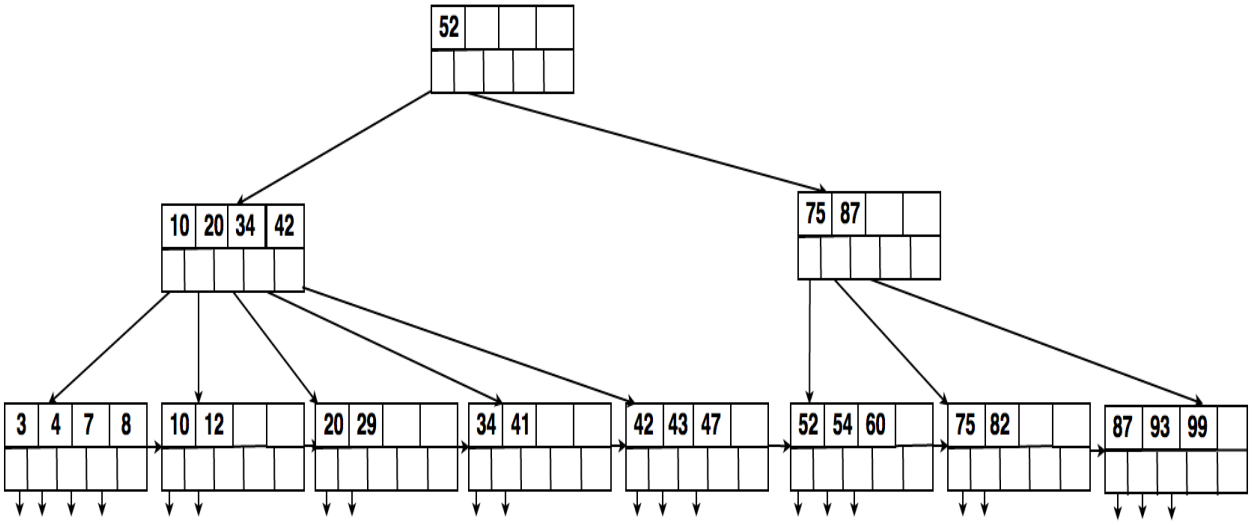
2. Suppose the index built is sparse and clustered (one key-pointer pair/block). How many blocks are required to store all the records of R (data file and index file)?
   Answer: The total number of index blocks + number of data blocks

$$(\text{one key per data block}/(4kB/8B)) + (1,000,000/(32kB/256B)) = 8002 \text{ Blocks}$$

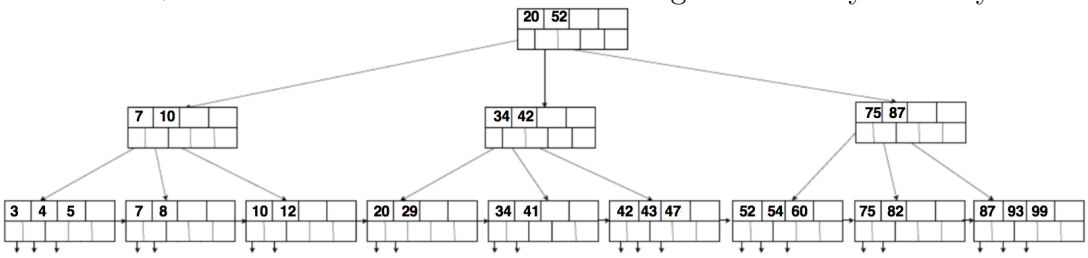
Note : This is not the only correct solution.

Question 2. B+ tree (30 points)

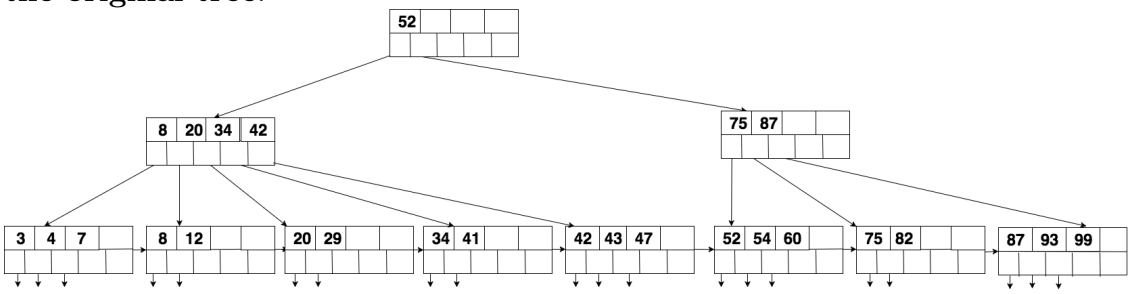
Consider the B+ tree of degree 2 (i.e., d = 2, which means each index node can hold at most 2d = 4 keys and 2d + 1 = 5 pointers) shown below:



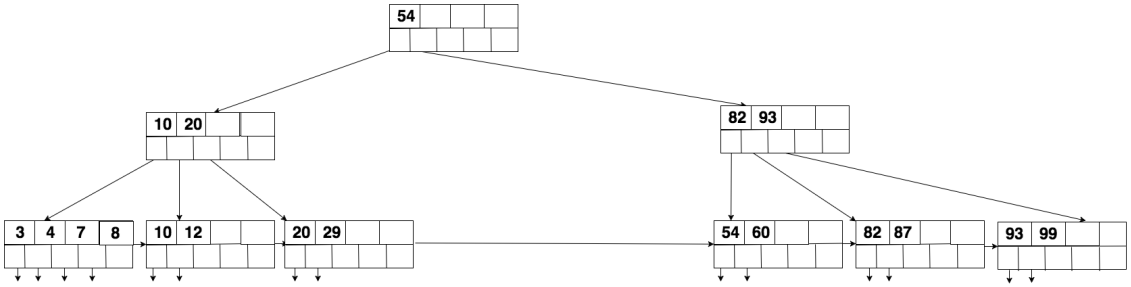
1. Draw the B+ tree that would result from inserting a data entry with key 5.



2. Show the B+ tree that would result from deleting the data entry with key 10 **from the original tree**.



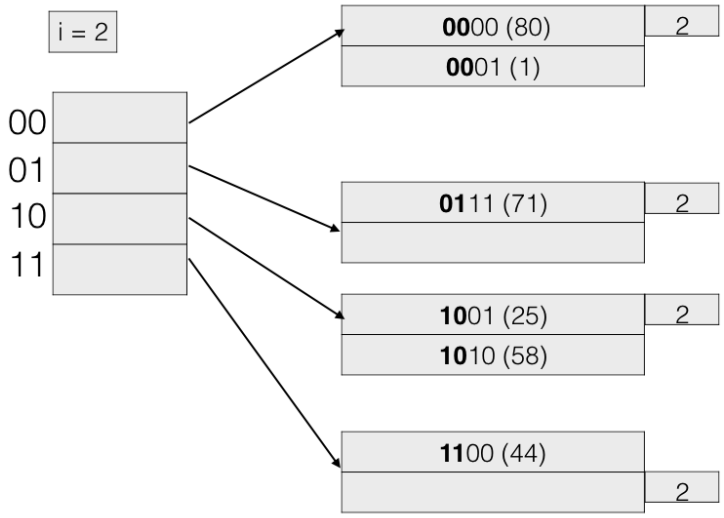
3. Show the B+ tree that would result from successively deleting the data entries with keys 34, 41, 42, 43, 47, 52 and 75 **from the original tree**.



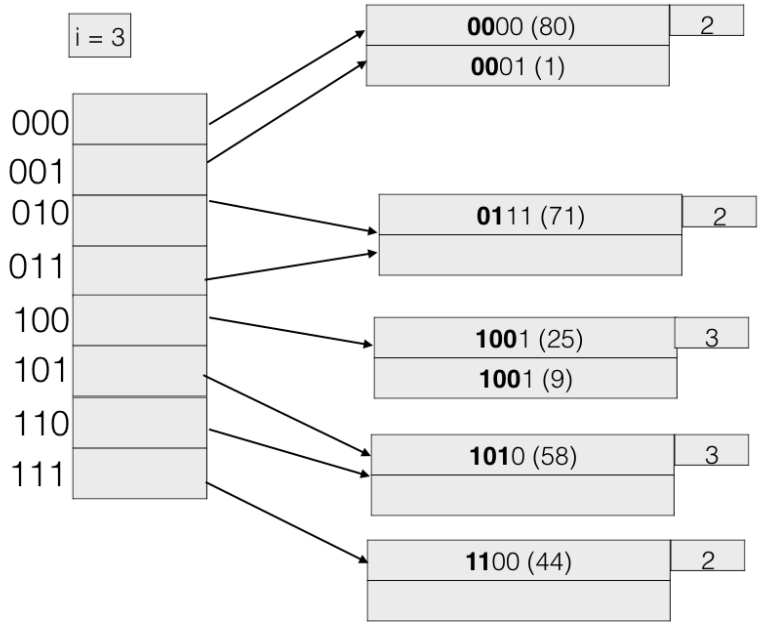
Question 3. Extensible Hashing

Consider you have been asked to index the following key values using an extensible hashing, in order: 1, 25, 44, 58, 71, 80. The hash function h(n) for key n is h(n) = n mod 16; i.e., the hash function is the remainder after the key value is divided by 16. Thus, the hash value is a 4-bit value. Assume that each bucket can hold 2 data items.

1. Draw the extensible hash table which obeys the above constraints.

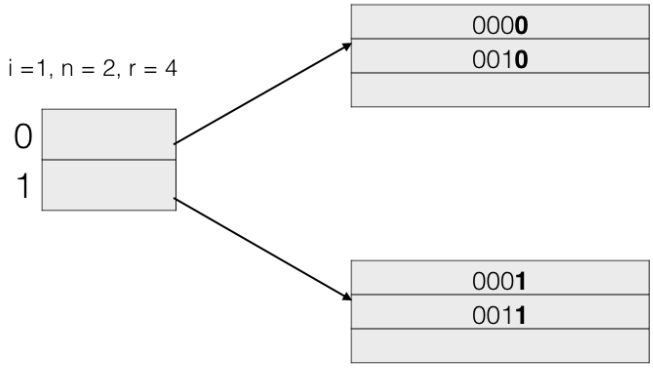


2. Now consider insertion of key 9 into the hash table, draw the resulting hash table.

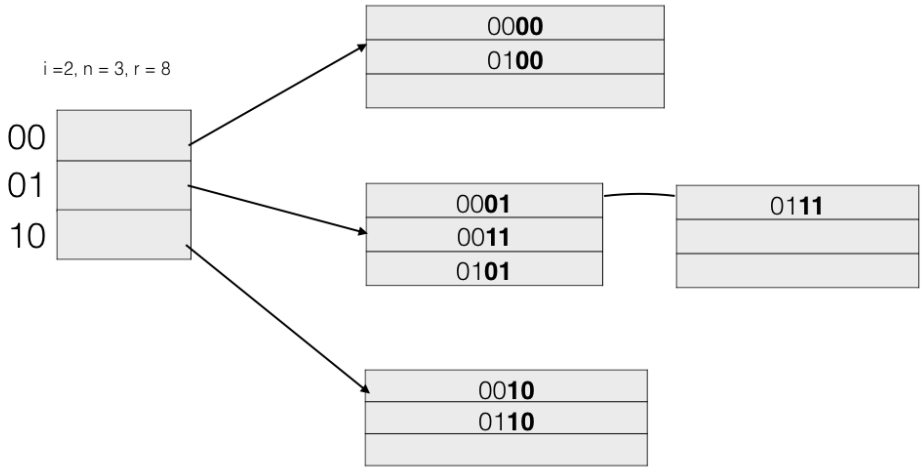


Question 4. Linear Hashing

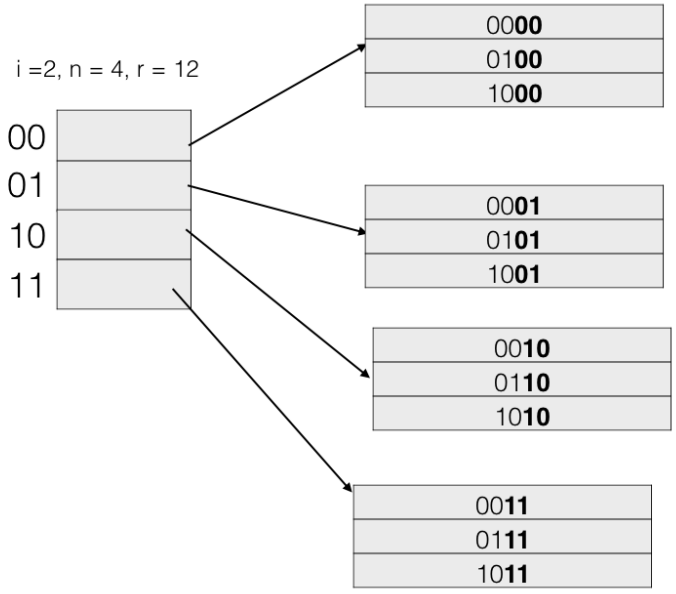
1. Insertion of 0000-0011



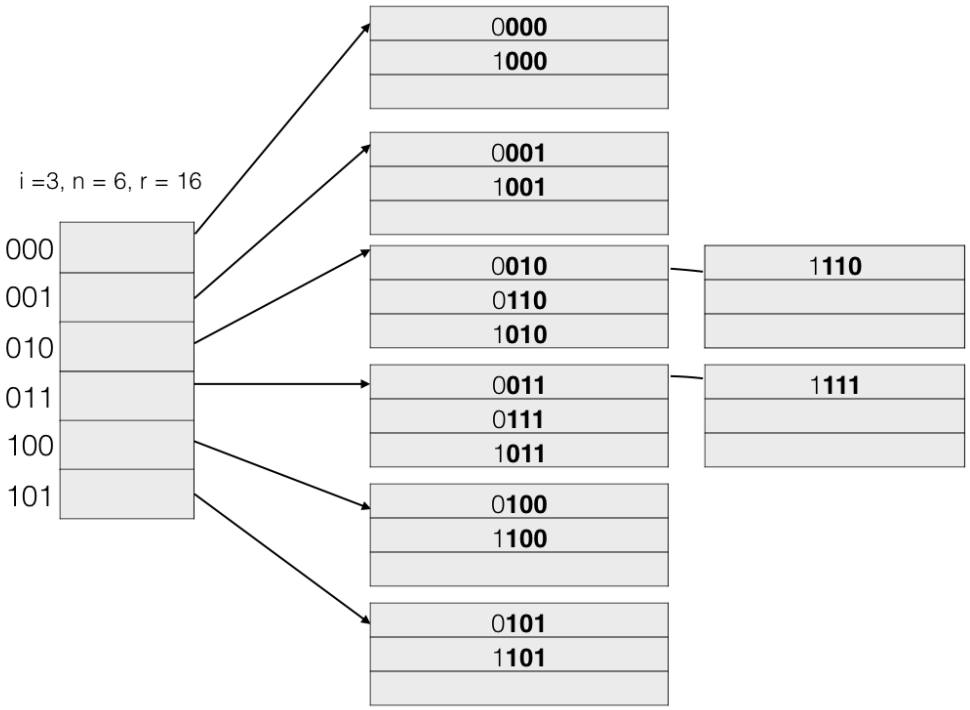
2. Insertion of 0100-0111



3. Insertion of 0101-1011



4. Insertion of 1011 -1111


i =3, n = 6, r = 16
000
001
010
011
100
101
0000
1000
0001
1001
0010
0110
1010
1110
0011
0111
1011
1111
0100
1100
0101
1101