Data Structures and Algorithms
CSYE 6205
Homework 6
Due: February 25, 2019

Put all your work, code, compiled, and executable files and documentation into a zip file named Homework6.zip and submit it via the drop box on the blackboard by the END of due date. Put your name on all code files. There will be a short quiz on this homework.

1. Consider this string : “abdceddfcabbeeeccefddaaaf”

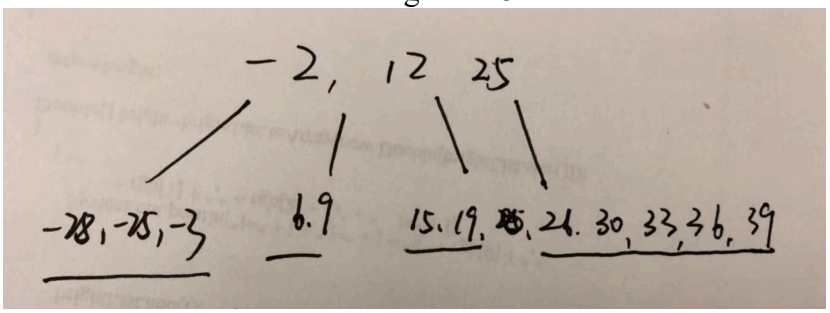
a) Use key-indexed counting sort algorithm to sort the string. Show each step and the results.
a1 b2 c3 d4 e5 f6
sort:
a b d c e d d f c a b b e e e c c e f d d a a a f → [1, 0, 0, 0, 0, 0]
a b d c e d d f c a b b e e e c c e f d d a a a f → [1, 1, 0, 0, 0, 0]
a b d c e d d f c a b b e e e c c e f d d a a a f → [1, 1, 0, 1, 0, 0]
a b d c e d d f c a b b e e e c c e f d d a a a f → [1, 1, 1, 1, 0, 0]
a b d c e d d f c a b b e e e c c e f d d a a a f → [1, 1, 1, 1, 1, 0]
a b d c e d d f c a b b e e e c c e f d d a a a f → [1, 1, 1, 3, 1, 0]
a b d c e d d f c a b b e e e c c e f d d a a a f → [1, 1, 1, 3, 1, 1]
a b d c e d d f c a b b e e e c c e f d d a a a f → [1, 1, 2, 3, 1, 1]
a b d c e d d f c a b b e e e c c e f d d a a a f → [2, 1, 2, 3, 1, 1]
a b d c e d d f c a b b e e e c c e f d d a a a f → [2, 3, 2, 3, 1, 1]
a b d c e d d f c a b b e e e c c e f d d a a a f → [2, 3, 2, 3, 4, 1]
a b d c e d d f c a b b e e e c c e f d d a a a f → [2, 3, 4, 3, 4, 1]
a b d c e d d f c a b b e e e c c e f d d a a a f → [2, 3, 4, 3, 5, 1]
a b d c e d d f c a b b e e e c c e f d d a a a f → [2, 3, 4, 3, 5, 2]
a b d c e d d f c a b b e e e c c e f d d a a a f → [2, 3, 4, 5, 5, 2]
a b d c e d d f c a b b e e e c c e f d d a a a f → [5, 3, 4, 5, 5, 2]
a b d c e d d f c a b b e e e c c e f d d a a a f → [5, 3, 4, 5, 5, 3]
[0, 3, 4, 5, 5, 3] → a a a a a
[0, 0, 4, 5, 5, 3] → a a a a a b b b
[0, 0, 0, 5, 5, 3] → a a a a a b b b c c c c
[0, 0, 0, 0, 5, 3] → a a a a a b b b c c c c d d d d d
[0, 0, 0, 0, 0, 3] → a a a a a b b b c c c c d d d d d e e e e e
[0, 0, 0, 0, 0, 0] → a a a a a b b b c c c c d d d d d e e e e e f f f

b) What is the running time complexity of the algorithm as compared to selection sort?
key indexed counting sort: O(n+9)
selection sort: O(n^2)

c) Write java code to sort the string using steps described in (a).

2. Consider Data: {9 30 -28 -25 -2 -3 6 12 15 19 25 26 33 36 39}

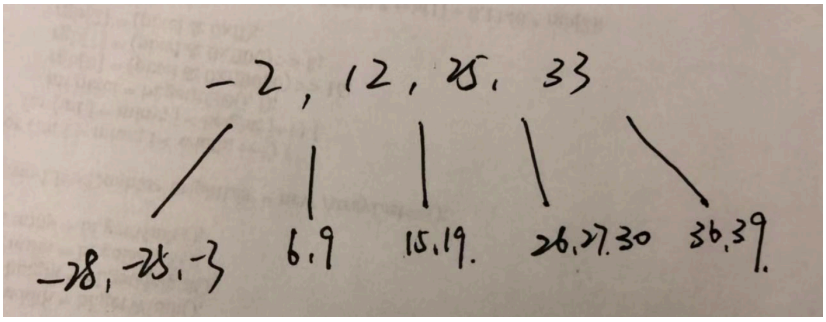
a) Build a btree with minimum degree of 3



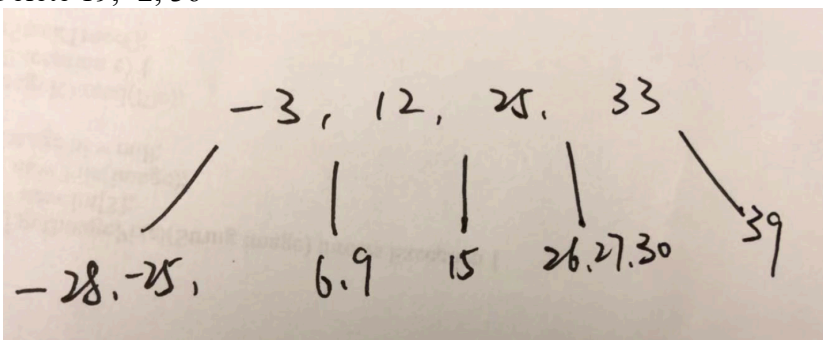
b) What is the order of this btree?
The order of this B-Tree is 6.

b) Why this is btree with minimum degree of 3?
Because that means the order should be 6. Or we can say the number of sub-nodes from a node is 6. Therefore, for each node, there must be 5 elements at most. For this tree,

d) Insert 27 into btree

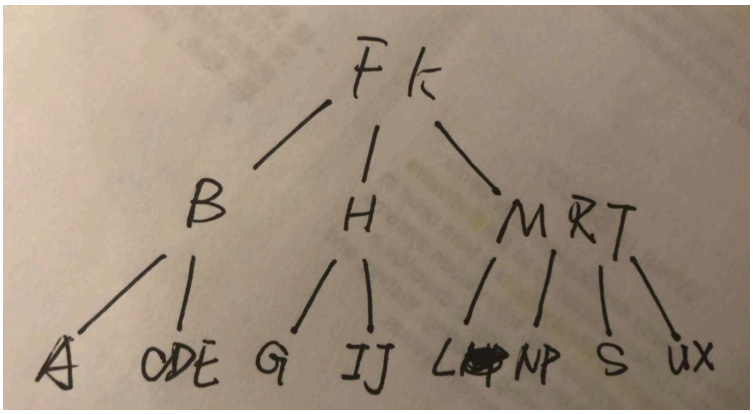


e) Delete 19, -2, 36



3. Consider the following sequence of letters:
'A'G'F'B'K'D'H'M'J'E'S'I'R'X'C'L'N'T'U'P'

a) Build BTree with order of t=4



b) What is minimum degree for this BTree?
The min-degree is 2.

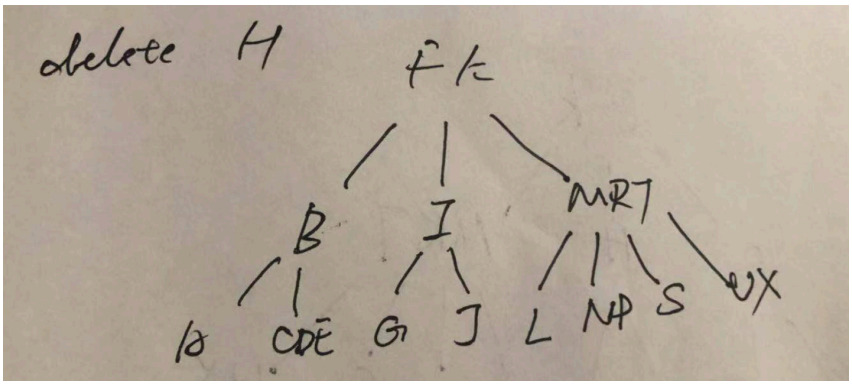
c) Write Java code to Insert into BTree

Check the .java file

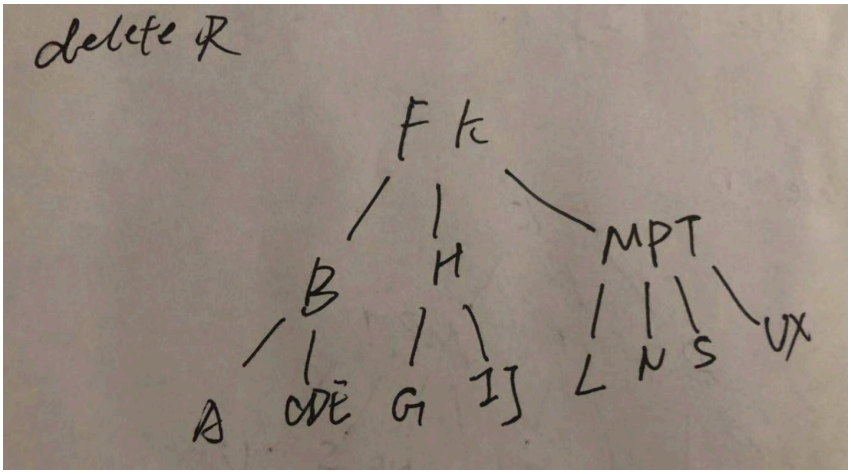
```
public void insert(T dValue) {
      if (find(dValue)==null) {
         Node<T> curNode = root;
         DataItem<T> tempItem = new DataItem<T>(dValue);
         while (true) {
            if (curNode.isFull())
            {
               split(curNode);
               curNode = curNode.getParent();
               curNode = getNextChild(curNode, dValue);
            } else if (curNode.isLeaf())
               break;
            else
               curNode = getNextChild(curNode, dValue);
         }
         curNode.insertItem(tempItem);
         size++;
      }
      else
         return;
   }
```

d) Consider 3-cases for deleting from B-tree,
Select delete element for each of 3-cases to delete from BTree in (a)

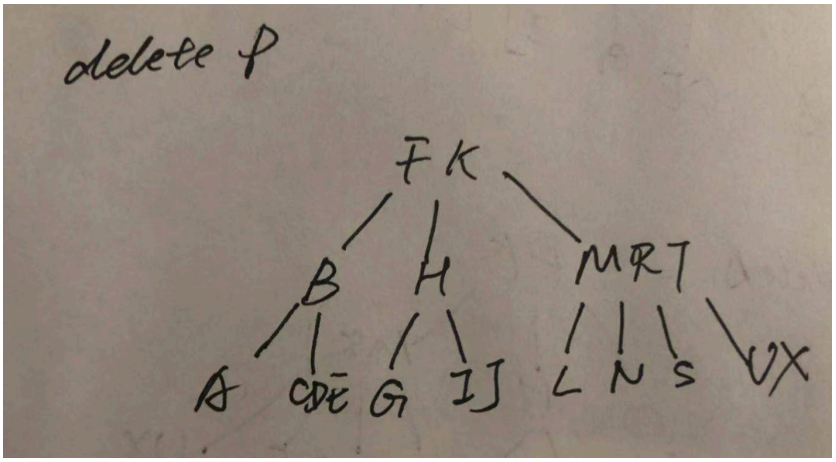
delete H

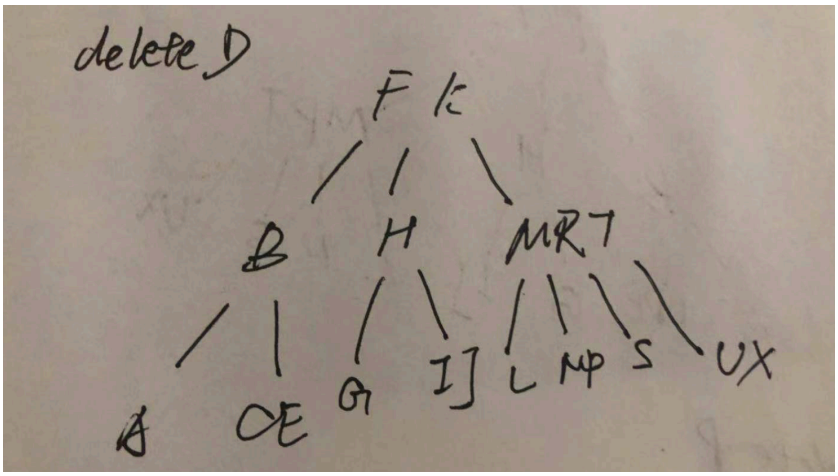


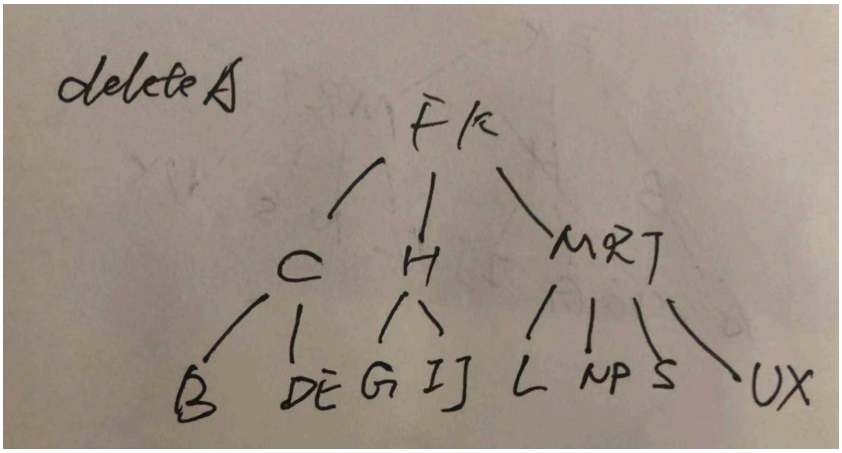
delete R



delete P







d) Write Java code for all 3 deletion cases
Check the .java file

e) Test Java code for (c) and (e) for the BTree you constructed in (a)
Check the .java file

H) Discuss height, time and space complexity
Height: 2
Time complexity: O(logn)
Space complexity: O(n)

4. Consider a disk with a sector size of 512 bytes, 1,000 tracks per surface, 100 sectors per track, five double-sided platters and a block/page size of 2,048 bytes. Suppose that the average seek time is 5 msec, the average rotational delay is 5 msec, and the transfer rate is 100 MB per second. Suppose that a file containing 1,000,000 records of 100 bytes each is to be stored on such a disk and that no record is allowed to span two blocks.

a) How many records fit onto a block?
2048/100=20

b) How many blocks are required to store the entire file?
1000000*100/2048=50000

Appendix-A DELETING from B-Tree

1. x is a leaf and contains the key (it will have at least t keys). This is

trivial - just delete the key.

2. x is an internal node and contains the key. There are 3 subcases:
   2a: predecessor child node has at least t keys
   2b: successor child node has at least t keys
   2c: neither predecessor nor successor child has t keys
3. x is an internal node, but doesn't contain the key. Find the child subtree of x that contains the key if it exists (call the child c). There are three subcases:
   3a: child c has at least t keys. Simply recurse to c.
   3b: child c has $t - 1$ keys and one of its siblings has t keys.
   3c: child c and both siblings have $t - 1$ keys.