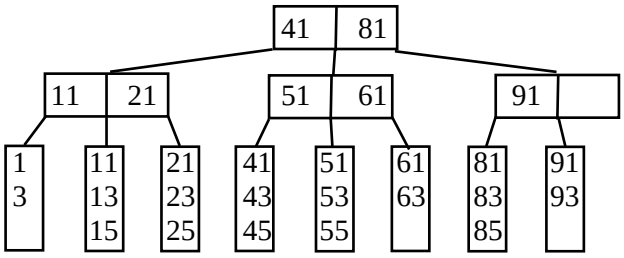
ECS 60 **Written Homework #3: Trees (20 points)** **Winter 2017**

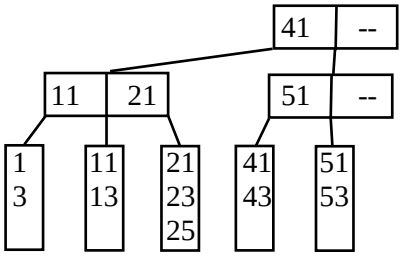
Due Friday, February 10th, 4:00 pm in 2131 Kemper

1. (4 points, 1 point each) For the following M = 3 and L = 3 BTree as the starting point for each part, draw the BTree that would result after the specified insertion. Sean's Rule for Insertion: Give left, give right, else split.



a. 62
b. 12
c. 24
d. 52

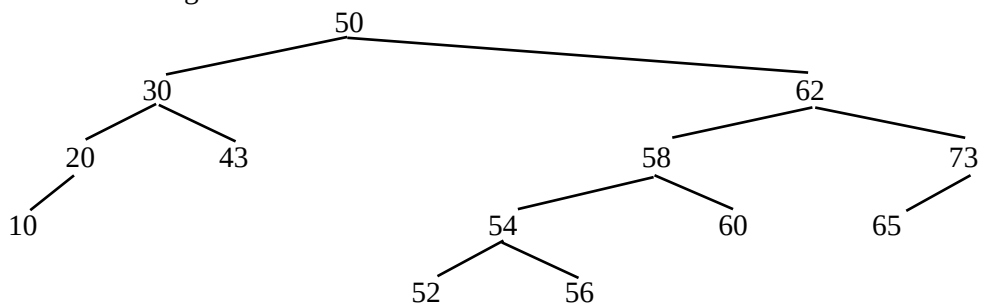
2. (4 points, 1 point each) For the following M = 3 and L = 3 BTree as the starting point for each part, draw the BTree that would result after the specified series of deletion. Sean's Rule for Deletion: Borrow left, merge left, borrow right, merge right.



a. 41
b. 25
c. 53
d. 11, 51

3. (1 point) Draw a binary search tree with only the elements {1, 2, 3, 4, 5, 6} in it, for which the postorder and inorder traversals generate the same sequence.

4. (4 points, 1 point each) Using the following binary search tree as the starting point for each part, draw the tree that results after the given operation. Sean's Rule for Deletions of nodes with two children in BSTs is to take the minimum of the right subtree.



a. Assuming it is an AVL tree, delete 43.
b. Assuming it is an AVL tree, insert 55
c. Assuming it is a splay tree, delete 54.
d. Assuming it is a splay tree, insert 5.

5. (2 points) Construct a Huffman trie based on the following distribution of letters. When combining two trees, always place the smaller on the left.

```
A
BB
CCCC
DDDDD
EEEEEE
FFFFFFFF
GGGGGGGGG
HHHHHHHHHH
JJJJJJJJJJJJ
KKKKKKKKKKKKK
LLLLLLLLLLLLLLLL
MMMMMMMMMMMMMMMMM
```

6. (2 points) Show that for any leaf *v* in a binary search tree, if *u* is the parent of *v*, then either key[*v*] is the largest key in the tree smaller than key[*u*], or key[*v*] is the smallest key in the tree larger than key[*u*]. (Heileman, p.216)

7. (2 points) Prove that the maximum number of nodes in a binary tree of height *h* is $2^{h+1} - 1$.

8. (1 point) In terms of *M* and *L*, what is the maximum height of a BTree with *n* elements?

Sources of questions:

Gregory L. Heileman, *Data Structures, Algorithms, and Object Oriented Programming*, New York, NY, McGraw-Hill, 1996.