

Worksheet for Trees

1. Given the **binary tree** operations, what tree or trees does the following sequence of statements produce?

```
Public void Ex11(){
    BinaryTree<Integer> t1 = new BinaryTree<Integer>(2);
    t1.attachLeft(5)

    BinaryTree<Integer> t2 = new BinaryTree<Integer>(8);
    t2.attachLeft(6);
    t2.attachRight(7);

    t1.attachRightSubtree(t2);

    BinaryTree<Integer> t3 = new BinaryTree<Integer>(7);
    t3.attachLeft(3);
    t3.attachRight(1);

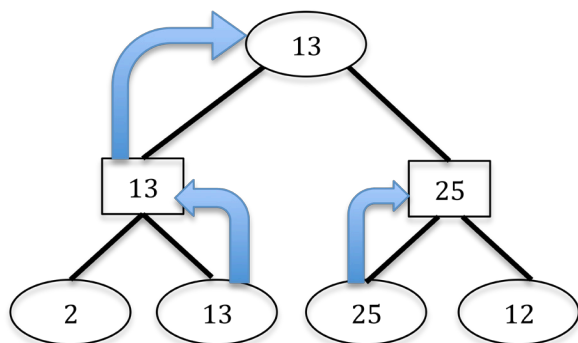
    BinaryTree<Integer> t4 = new BinaryTree<Integer> (1, t1, t3);
}
```

2. Consider a nonempty binary tree with two types of nodes: **min** nodes and **max** nodes. Each node has an integer value initially associated with it. You can define the value of such a *minimax* tree as follow;

If the root is a min node, the value of the tree is equal to the minimum of

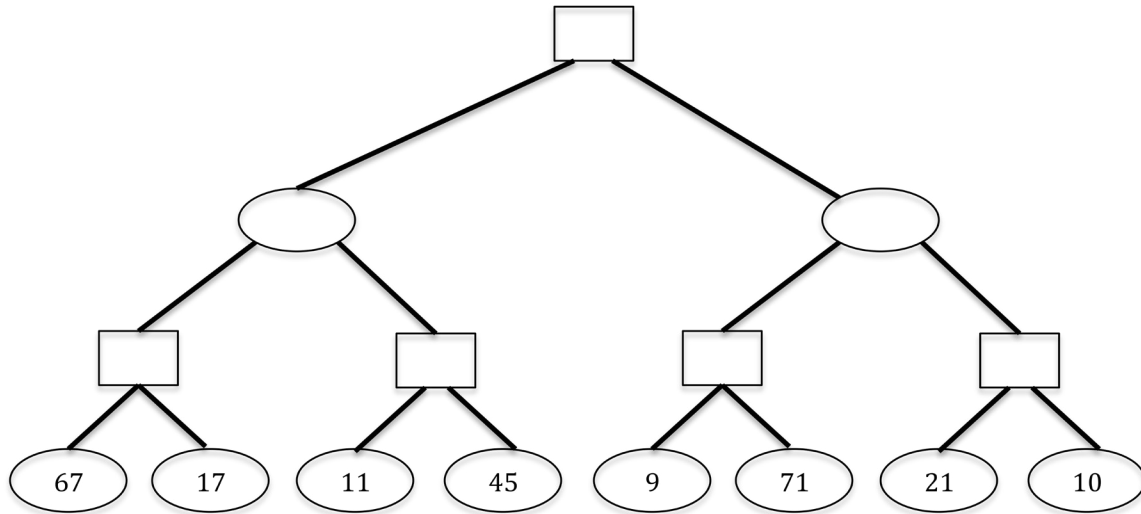
- The integer stored in the root
- The value of the left subtree, but only if it is nonempty
- The value of the right subtree, but only if it is nonempty

If the root is a max node, the value of the tree is equal to the maximum of the above three values.



An example of the minmax tree (square node: max node, circle node: min node)

Compute the value of the minmax tree in below tree. Each node is labeled with its initial value. (square: max node, circle: min node)



3. A binary tree is **strictly binary** if every non-leaf node has exactly two children. Prove by induction on the number of leaves that a strictly binary tree with n leaves has exactly $2n - 1$ nodes.

3. Execute the following sequence of operations on an initially empty ADT (abstract data type) implemented as

- A binary search tree
- A 2-3 tree
- A 2-3-4 tree
- A red-black tree
- An AVL tree

And show the underlying tree after each operation.

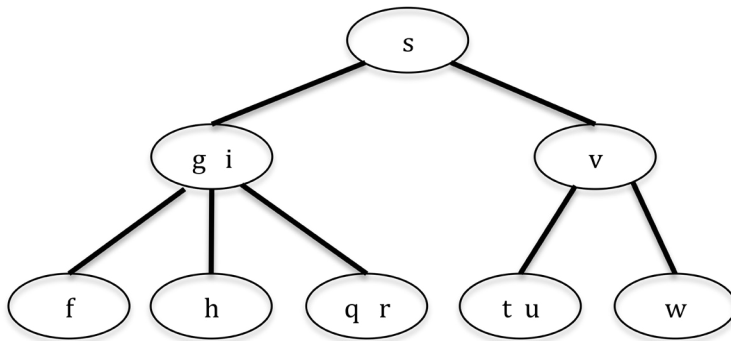
```

t.treeInsert(17)
t.treeInsert(78)
t.treeInsert(20)
t.treeInsert(57)
t.treeInsert(51)
t.treeDelete(17)
t.treeInsert(60)
t.treeInsert(70)
    
```

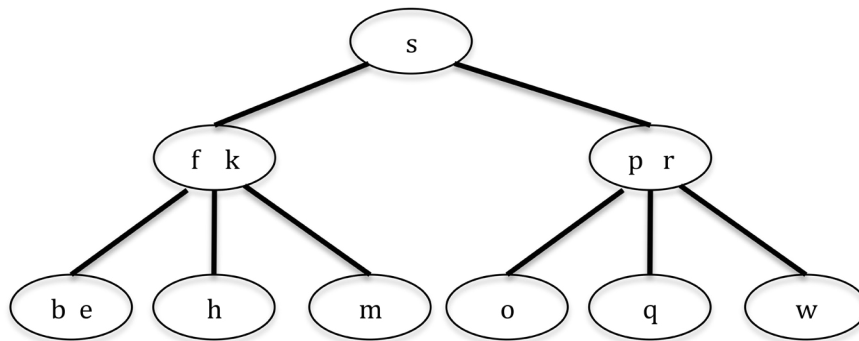
```
t.treeInsert(40)
t.treeDelete(57)
t.treeInsert(90)
t.treeInsert(19)
t.treeDelete(20)
t.treeDelete(70)
```

4. What are the advantages of implementing tree with a 2-3 tree instead of a binary search tree? Why do you not, in general, maintain a completely balanced binary search tree?

5. Given the following 2-3 tree, draw the tree that results after inserting m,a,d,x, and z into the tree.

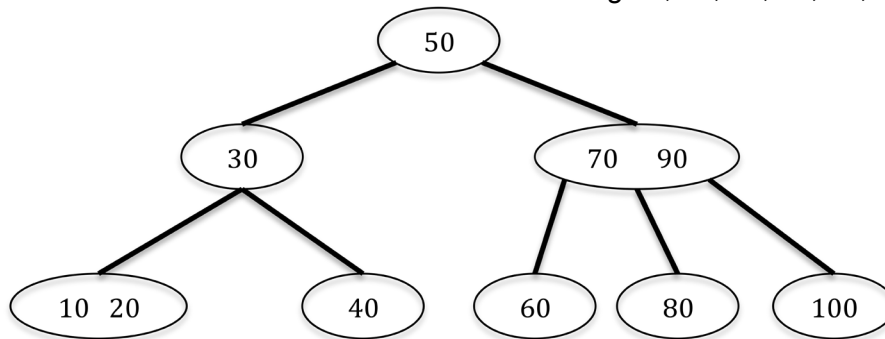


6. Given the following 2-3 tree, draw the tree that results after removing r,e,g, and b from the tree.

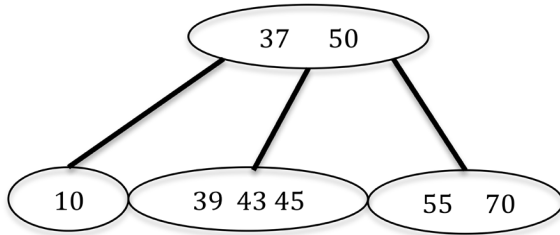


7. Draw the 2-3-4 tree that results from inserting p,c,m,j,v,f, and b, in the order given, into a 2-3-4 tree that contains a single node whose value is n.

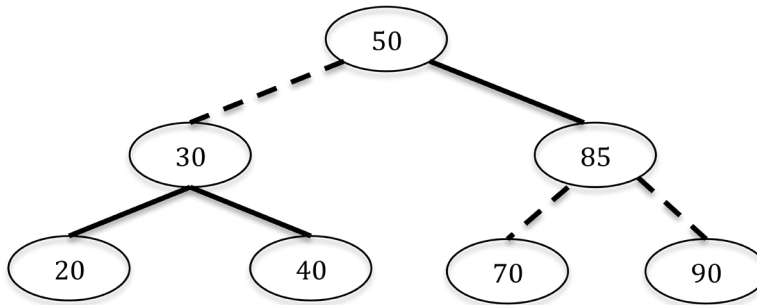
8. Draw the 2-3-4 tree that results from inserting 39, 38, 37, 36, 35, 34, 33, and 32 into it.



9. Change the following 2-3-4 tree to red-black tree.



10. Change the following red-black tree to a 2-3-4 tree.



10. Why does a node in a red-black tree require less memory than a node in a 2-3-4 tree?

11. Given a B-tree of degree 5 and a height of 3;

- a. What is the maximum number of nodes (including the foot)?
- b. What is the maximum number of records that can be stored?

12. The B-tree traversal algorithm presented in this class assumes that internal memory is large enough to accommodate the recursive stack that contains up to h blocks, where h is the height of the B-tree. If you are in an environment where this assumption is not true, modify the traversal algorithm so that the recursive stack contains block numbers rather than the actual blocks. How many block accesses does your algorithm have to perform?