CSCD359 Homework1, Winter 2012, Eastern Washington University. Cheney, Washington.

Name: Eric Fode **EWU ID:00530214**

Solution for Problem 1

Idea: Go through the list checking to see if the next item is less then the current node, if it is add it to the left then call this function on the new left node save the return value of this recursive call to be used as the index at which the right node gets added from, then call the function on the right node finally return the current position in the array.

preOrderTree($treeArray, root, index$**)**

Input: A preordered array,The root of the tree (starting with a node that has the first value of the array,the arrayIndex

Result: The tree that the preorder array represent in attached to the root node

```
1 begin
2     if treeArray[index] < root.value then
3         root.left = node(treeArray[index]));
4         index ⟵ preOrderTree(treeArray, root.left, index);
5         root.right = node(treeArray[index]));
6         index ⟵ preOrderTree(treeArray, root.right, index);
7     return index + 1;
```

Analysis: This should take $O(n)$ time

Solution for Problem 2

Idea: Find the mid point of the array make it the root of the current tree, recurse for the left and right sides of the tree using only the upper half or the lower half of the array until there are no items left

sortedTree($treeArray$**)**

Input: A slice of a sorted array

Result: A BST that has a height bounded by $log(n)$

```
1 begin
2     if treeArray.size = 1 then
3         return newnode(treeArray[0]);
4     root = newnode(treeArray[treeArray.size/2]);
5     root.left ⟵ sortedTree(treeArray[0 : treeArray.size/2]);
6     root.right ⟵ sortedTree(treeArray[treeArray.size/2 + 1 : treeArray.size]);
7     return root;
```

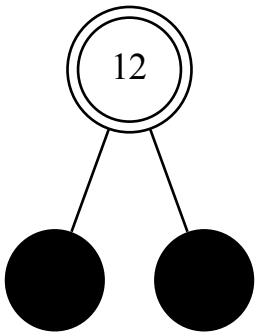
Analysis: This should take $O(n)$ time

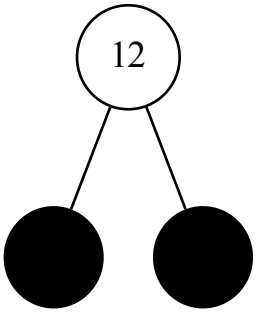
Solution for Problem 3

12

CSCD359 Homework1, Winter 2012, Eastern Washington University. Cheney, Washington.

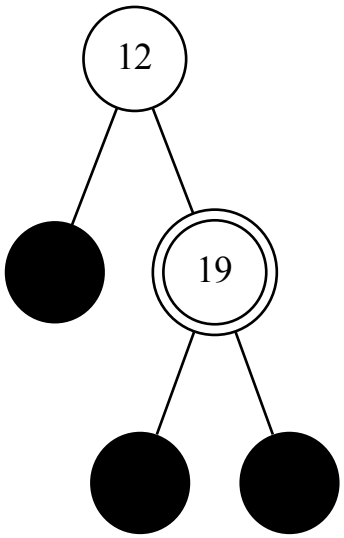
Name: Eric Fode **EWU ID:00530214**





19

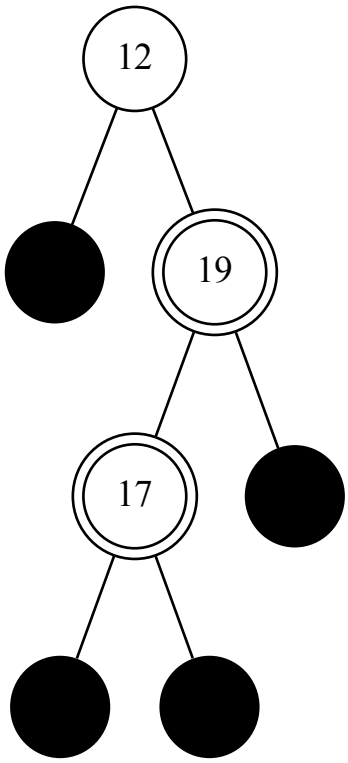
Name: Eric Fode **EWU ID:00530214**



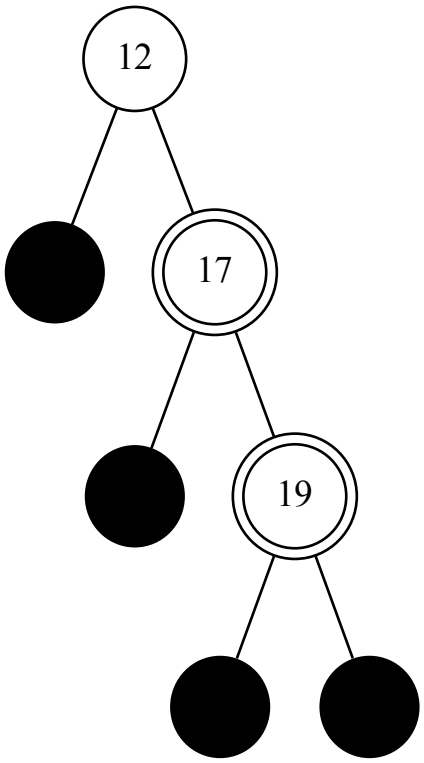
17

CSCD359 Homework1, Winter 2012, Eastern Washington University. Cheney, Washington.

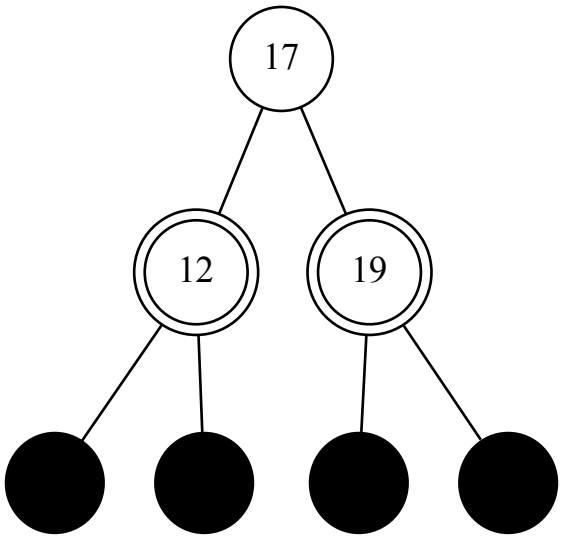
Name: Eric Fode **EWU ID:00530214**



Name: Eric Fode **EWU ID:00530214**

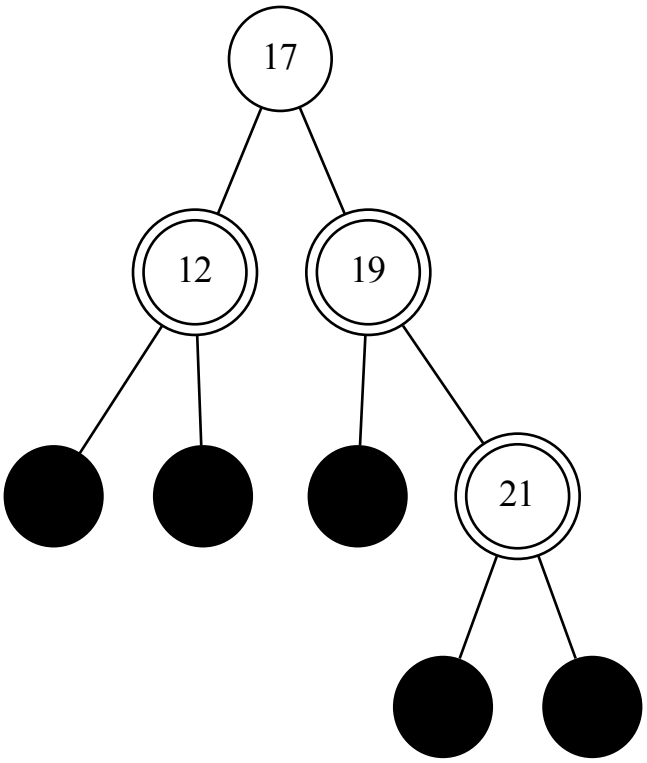


Name: Eric Fode **EWU ID:00530214**

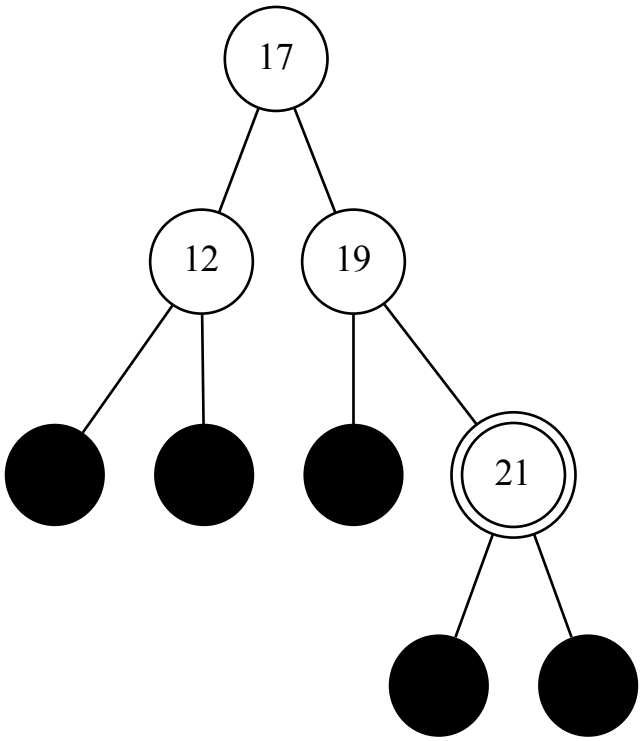


21

Name: Eric Fode **EWU ID:00530214**

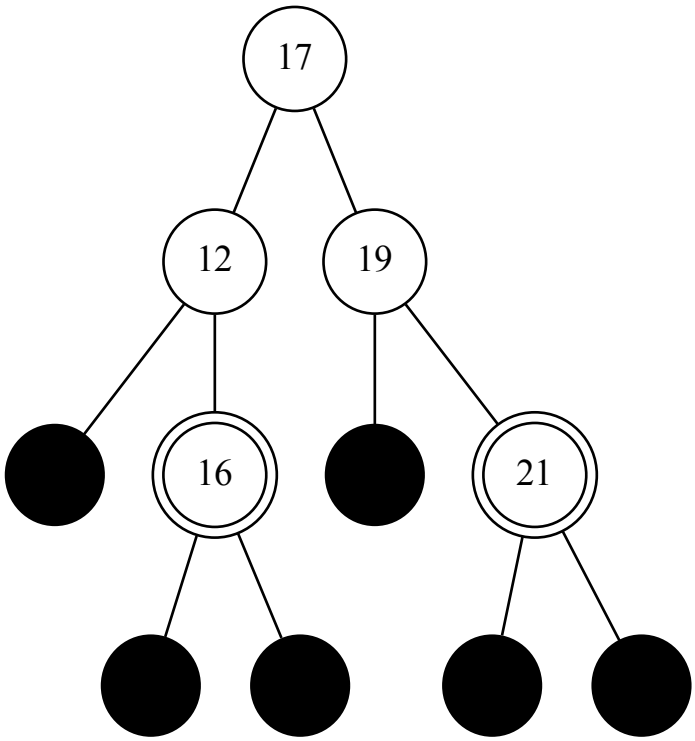


Name: Eric Fode **EWU ID:00530214**



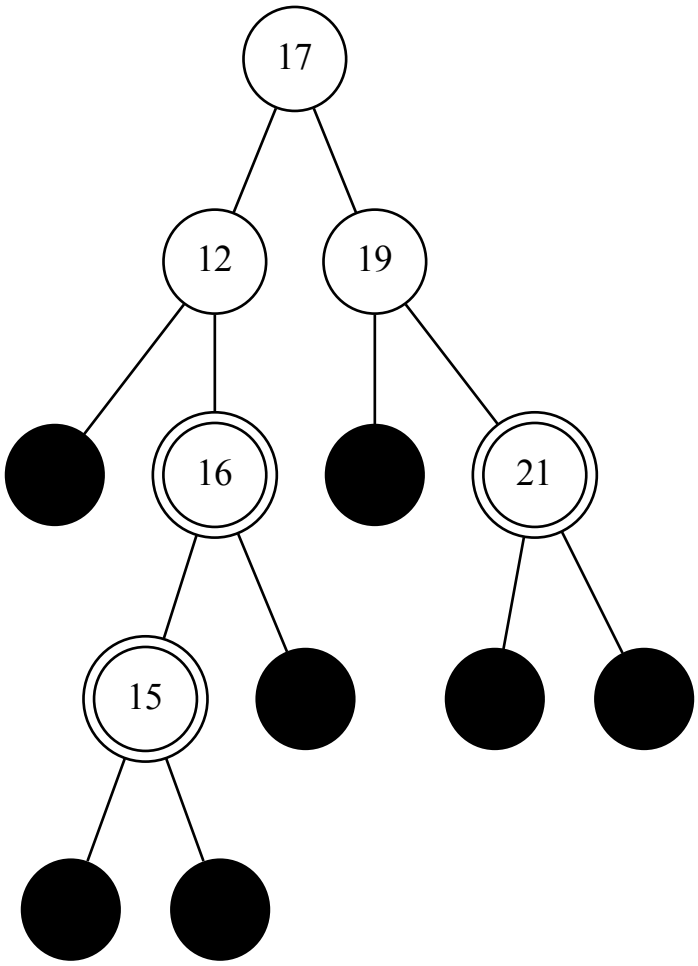
16

Name: Eric Fode **EWU ID:00530214**



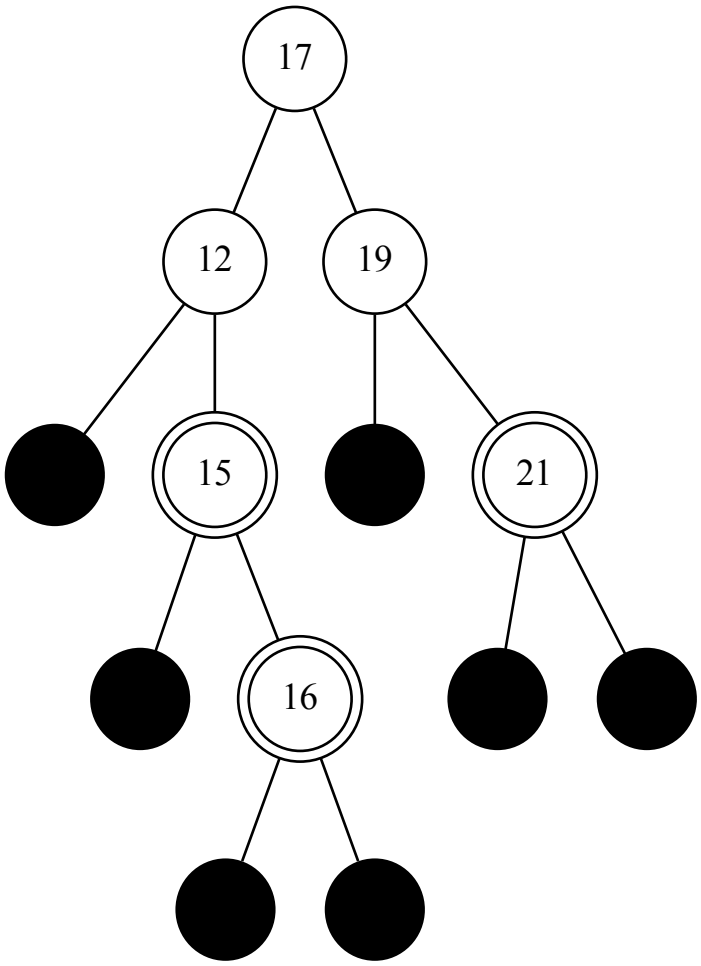
15

Name: Eric Fode **EWU ID:00530214**



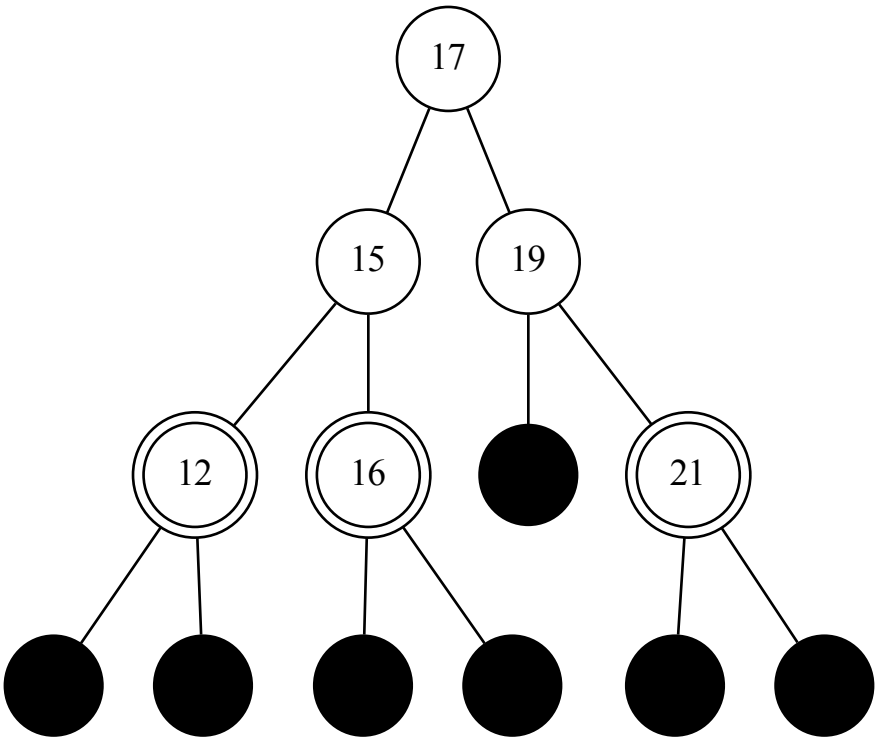
CSCD359 Homework1, Winter 2012, Eastern Washington University. Cheney, Washington.

Name: Eric Fode **EWU ID:00530214**



CSCD359 Homework1, Winter 2012, Eastern Washington University. Cheney, Washington.

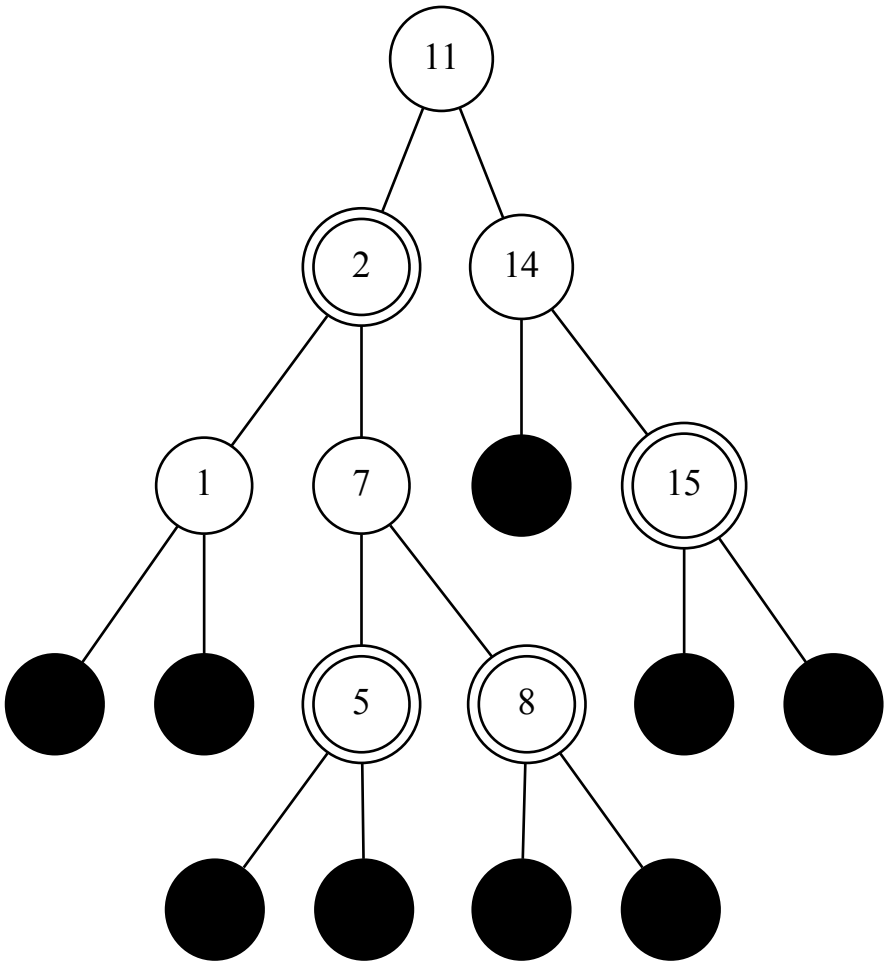
Name: Eric Fode **EWU ID:00530214**



Solution for Problem 4

CSCD359 Homework1, Winter 2012, Eastern Washington University. Cheney, Washington.

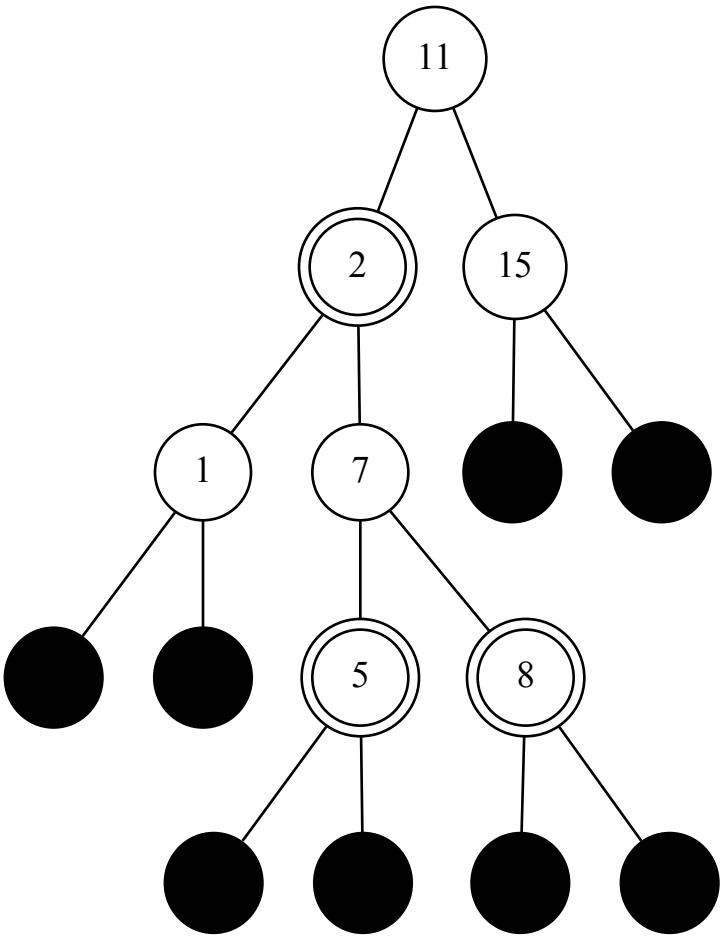
Name: Eric Fode **EWU ID:00530214**



14

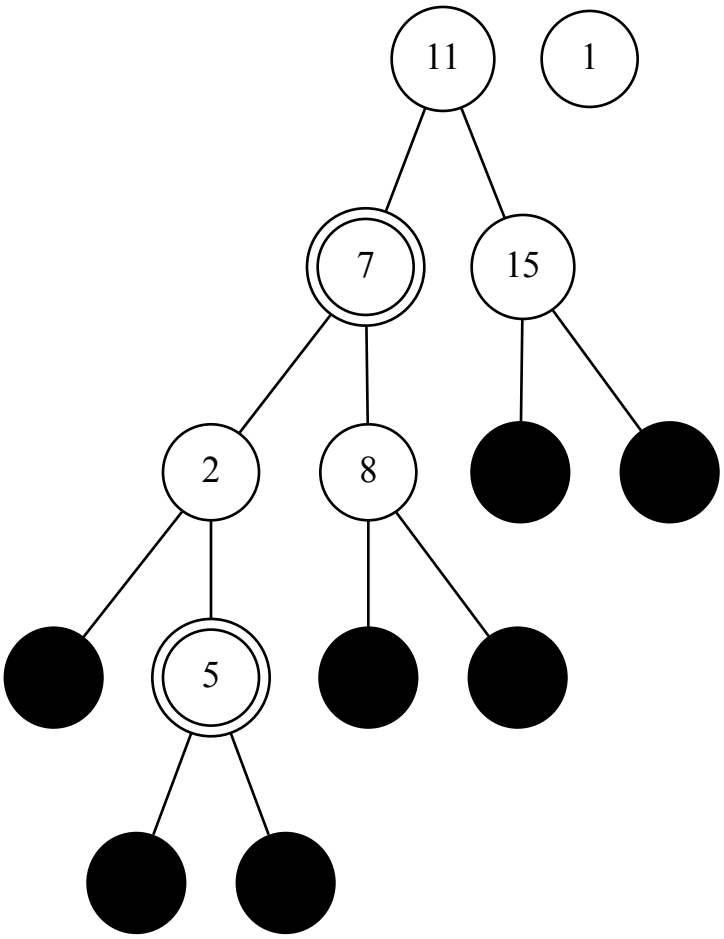
CSCD359 Homework1, Winter 2012, Eastern Washington University. Cheney, Washington.

Name: Eric Fode **EWU ID:00530214**



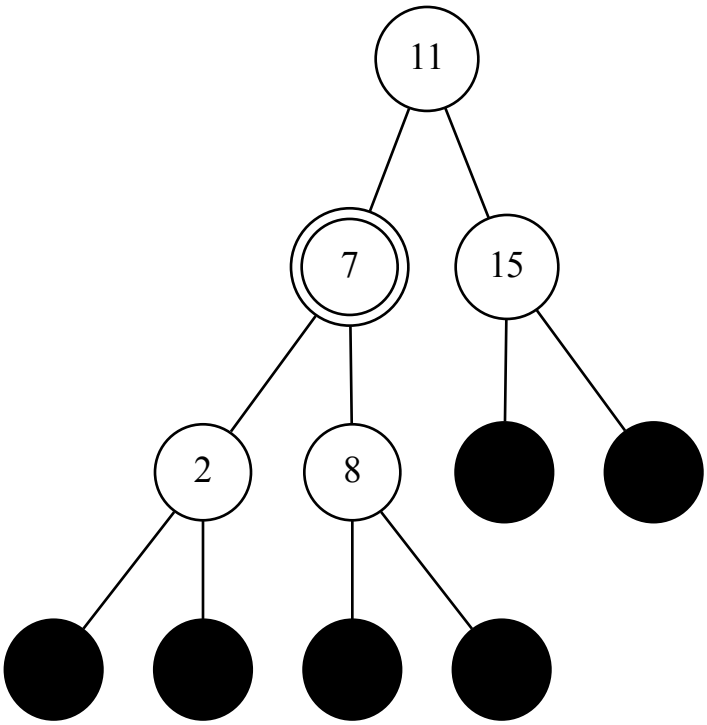
1

Name: Eric Fode **EWU ID:00530214**



5

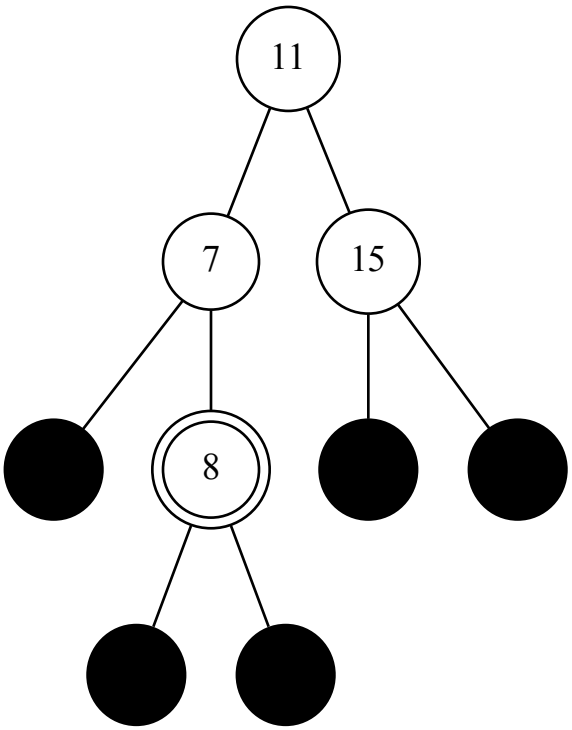
Name: Eric Fode **EWU ID:00530214**



2

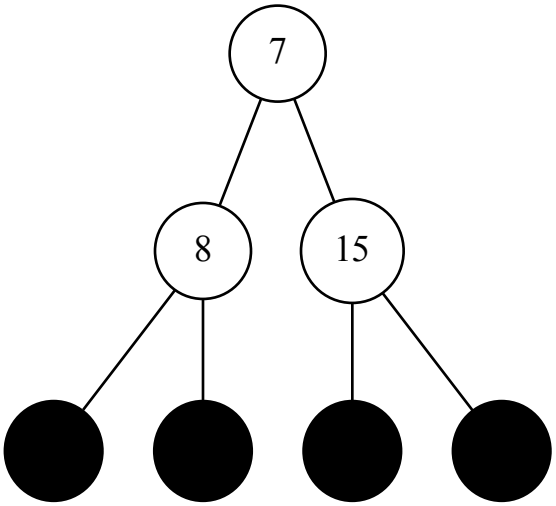
CSCD359 Homework1, Winter 2012, Eastern Washington University. Cheney, Washington.

Name: Eric Fode **EWU ID:00530214**



11

Name: Eric Fode **EWU ID:00530214**



The rest of the step are basic bst pruning and do not need to be shown becuase they don't use any of the Red-Black rules