Alexander Shah

Homework 6: Trees

EN.605.202.81 Section 84

1.
For each level of a binary tree, there can be at most two nodes per parent. At the root, there are no ancestor nodes. For each level of the tree we can expect the tree to double. At level 0 we have the root node. Level 1 has 2 nodes and 1 ancestor, the root. Level 2 has 4 nodes and 1 new ancestor totalling 2, level 3 has 8 nodes and 1 new ancestor totalling 3. So for each level, we have one new ancestor. This follows the pattern that the number of ancestors is also n.

2.
At the root, there is one node, at level 1, there are 2 nodes plus the root so 3, at level 2 there are 4 nodes plus the previous 3 so 7, at level 3 there are 8 nodes plus the previous 7 so 15 total. E.g. the number of leaves, n, at level 3 is 8 and the number of previous nodes is 2n-1, 15. This shows that for n leaves, the number of nodes in a binary tree is 2n-1.

3.
Using pointer fields in an m-ary tree, each parent can have up to m children, where the parent points to the children. If the parent node has less than m children the missing pointers will be null. In the case of leaf nodes, there are no pointers. If there are n nodes, there will be n*m pointer fields and since the root doesn't have a parent, there will be n-1 pointers. If we subtract the used pointers (n-1) from the total number of pointers (n*m) we get $(n * m) – (n – 1) \rightarrow nm – n + 1 \rightarrow n(m – 1) + 1$. So for n nodes there are n(m-1)+1 null child pointers.

4.
```
Method fib_tree(n):
    if n==0 or n==1: return new Node(n)
    else:
        root = new Node(n)
        root.left = fib_tree(n-1)
        root.right = fib_tree(n-2)
        return root
```

This method establishes a base case when n is 0 or 1, otherwise it recursively creates subtrees to the left and right of the root node.

5.
a.
Yes, the subtrees contain exactly 1 node in the base case or follow the binary tree structure by only containing up to 2 nodes per parent.

b.
For 0 or 1 there is only one leaf, for a tree of order n the number of leaves is also n because every value from n to 1 or 0 will be added to the tree.

c.
For 0 or 1 there is only one node. For a tree of order n the number of levels will be the same number n, since the tree grows linearly based on n.