Hannah Kerr and Emma Steinman
CS 271 Project 6
December 17, 2017

1.a To prove that a complete binary tree with height h will have $2^{h+1} - 1$ total nodes we will start with a base case where $h = 0$. A complete binary tree with height zero is just the root, which has $2^{0+1} - 1 = 1$ total nodes. Assume that all complete binary trees with heigh k, where $0 \leq k \leq h - 1$, have $2^{h+1} - 1$ total nodes. Now assume we have a complete binary tree with height h. Each of the subtrees of the root have height $h - 1$. Since this is in the range of k, we can apply the inductive hypothesis, so each subtree has $2^{(h-1)+1} - 1 = 2^h - 1$ total nodes. Thus there are $2(2^h - 1) + 1 = 2^{h+1} - 1$ total nodes in the complete binary tree with height h.

1.b To prove that a complete binary tree with n nodes will have $\frac{n-1}{2}$ internal nodes, we start with a base case where $n = 1$. A complete binary tree with one node is just the root, which is not an internal node if it has no children. Thus the complete binary tree has $\frac{1-1}{2} = 0$ internal nodes. Assume this is true for all complete binary trees with k nodes, where $0 \leq k \leq n - 1$. Now assume we have a complete binary tree with n nodes. Each of the subrees will have $\frac{n-1}{2}$ nodes and since this is in the range of k we can apply the inductive hypothesis. Thus each subtree will have $\frac{\frac{n-1}{2}-1}{2} = \frac{n-1-2}{4} = \frac{n-3}{4}$ internal nodes and the entire tree will have $2(\frac{n-3}{4})+1 = \frac{n-3}{2}+1 = \frac{n-3+2}{2} = \frac{n-1}{2}$ internal nodes in the complete binary tree with n nodes.

2 For a node x with an empty right subtree, to find the successor, y you find the ancestor up to the left as far as possible and then one up to the right. This, y will be the lowest ancestor of x whose left child is also an ancestor of x because the left child of the successor is as far up and to the left you can travel from x. The successor, y, is the lowest ancestor of x whose left child is also an ancestor of x because the left child of the left child of the successor will not be on the path travelling up and to the left from x.
If x is the child of its successor y, it still upholds the properties that the successor is the lowest ancestor of x whose left child is also an ancestor of x because every node is an ancestor to itself.
Alternatively, if y is the successor of x, x is the predecessor of y. To find the predecessor when the left subtree exists, you find the greatest item in the left subtree. x will be in the left subtree because it exists and the predecessor will be less than the successor. Once in the left subtree, travelling all the way to the right will give you x because as the predecessor of y, x is the greatest element in the left subtree.