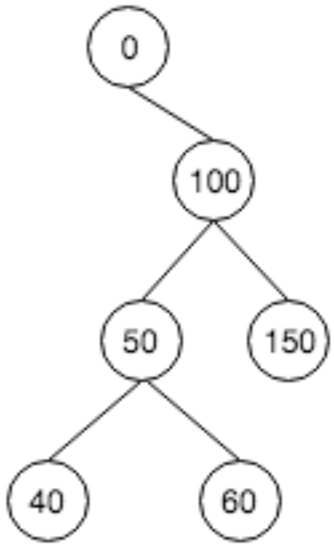
Zijian Gao
26319560
2/29/2016
CS311 HW3

12.2-4.)



Search is for k=60, A = {40} B = {0, 100, 50, 60} C={150}. $40 \in A$ and $0 \in B$ but 40 > 0, which violates the professor's claim.

12.2-5.)

Show: If a node in a binary search tree has two children, then its successor has no left child and its predecessor has no right child.

Prove by contradiction

Successor has no left child:
Assume the successor of some arbitrary node in a binary search tree has a left child. The successor is defined as the smallest key that is greater than the node. This is either the leftmost key of the right subtree, or the first parent node that has a left child that contains a path to the node. Since it is assumed that the node's successor has a left child, and given that the node has two children, the successor will be contained in the node's right subtree. This is a contradiction because if the successor has a child, then the successor is not the successor as its left child would in fact be the successor, so it cannot have a left child.

Predecessor has no right child:
Assume the predecessor of some arbitrary node in a BST has a right child. Similar to the successor, in this case where the target node has a left child node, the predecessor must be the maximum node in the left child subtree of the target node. If the predecessor has a right child, then there must exist another node that is greater than the "predecessor" but less than the target node, which would mean that the predecessor is not the predecessor creating a contradiction, so the predecessor cannot have a right child.