1. a) The tree's root is the only node in the tree without a parent from which all the nodes in the tree are descended from; therefore, the root of the tree is the node containing the number 60.
b) 60 is a parent, 20 is a parent, and 40 is a parent, because they all have nodes "underneath" them (i.e. children).
c) 20, 70, 10, 40, 30, and 50 are all of the children from all the parents in part b. This is also every node in the tree, excluding the root.
d) 70 is the sibling of 20, since they share the same parent node. 10 is also the sibling of 40, as well as 30 and 50.
e) 40, 20, and 60 are all the ancestors of 50, because they all lie on the path from the root of the tree (60) to 50.
f) The descendants of 20 are the nodes that lie on a path from 20 to a leaf. Thus, 10, 40, 30, and 50 are all the descendants of 20.
g) The leaves in the diagram are the nodes that have no descendants. 70, 10, 30, and 50 are all leaves.

2. The height of the tree in the diagram is the path from the root to a leaf in the tree that contains the largest number of nodes. Thus, the height of the tree is 4, following the path 60-20-40-50, or, alternatively, 60-20-40-30.

4. The preorder traversal of a binary tree examines every root node of a subtree just as it sees it; then, it examines left subtrees of a given root node first before going to the right subtrees. The traversal is therefore in the order 60, 20, 10, 40, 30, 50, 70.

The inorder traversal of a binary tree examines every node the second time it sees it. 10, 20, 30, 40, 50, 60, 70 is the order.

The postorder traversal of a binary tree examines every node when it will be the last time it sees it. The order is 10, 30, 50, 40, 20, 70, 60.

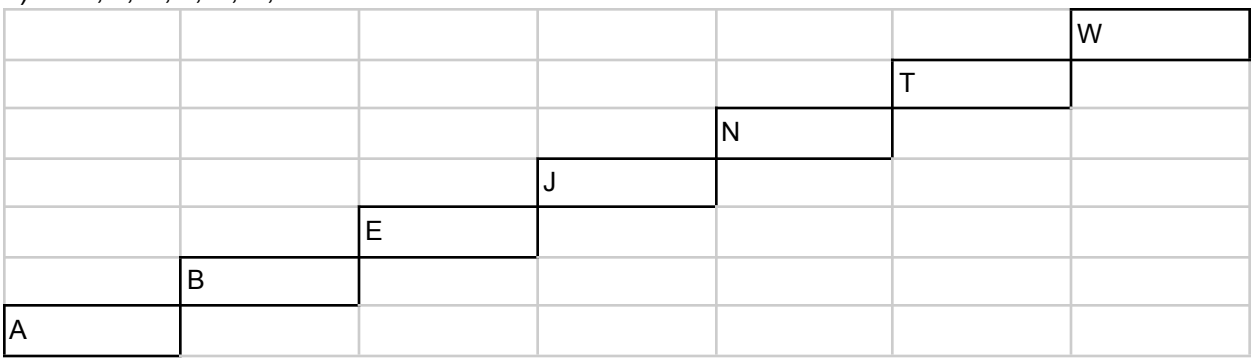
5. a) The method isLeaf, which checks if a binary tree is a one node tree, would need to check if the data member that represents the root of the tree is not nullptr (i.e it contains something), and that root's left and right subtrees are nullptr, meaning that the tree is a one-node tree that is only a leaf without children.
b) This could not be implemented by a client, assuming private members and encapsulation is upheld, preventing the clients from directly accessing the "root" of the binary tree object.

6. With a binary search tree, the figure would be produced by inserting in the order of the preorder traversal. 60, 20, 10, 40, 30, 50, 70 is the order.

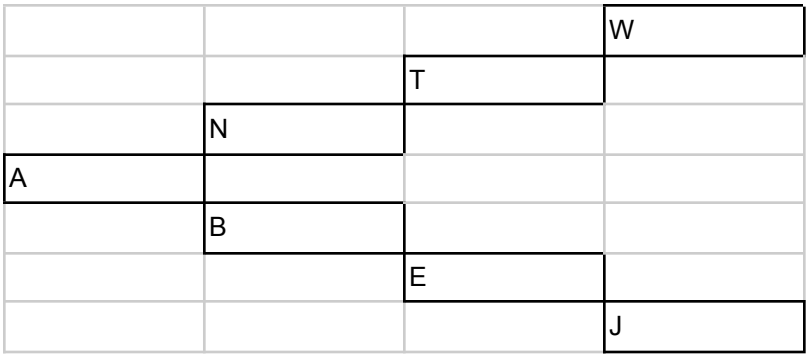
7) a. 60-20-40-30.
b. 60-20-10.

8) The figure is a binary search tree. For a given node “root”, every node in its left subtree is less than it, and every node in its right subtree is greater than it. Therefore, it is a binary search tree.

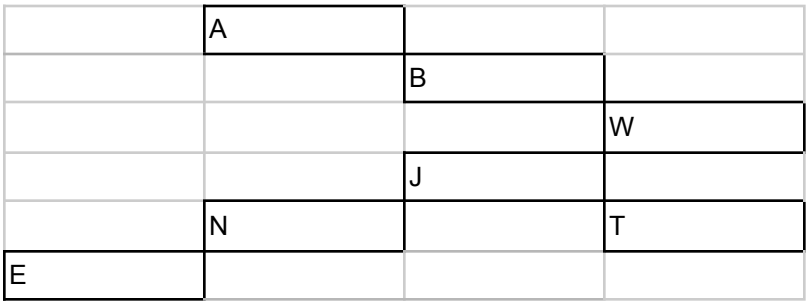
9) a. W, T, N, J, E, B, A



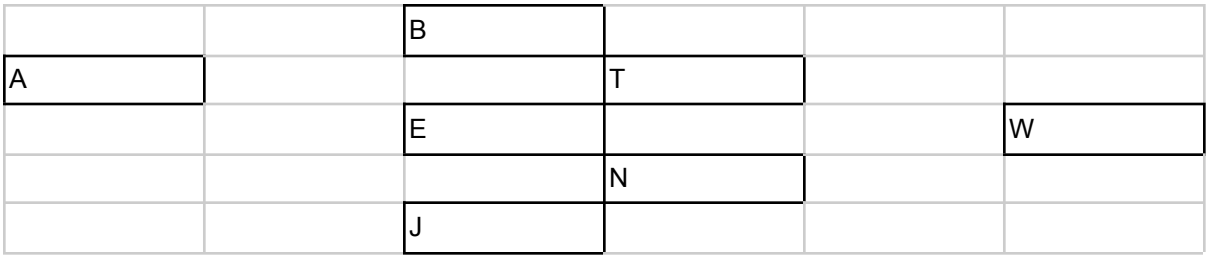

b. W, T, N, A, B, E, J




c. A, B, W, J, N, T, E




d. B, T, E, A, N, W, J



10) a. 9 contains the value that comes immediately after the root, because it is the smallest number in the right subtree (i.e., the smallest number that is greater than the root, so it is immediately after the root).
b. 4, 2, 8, 5, 1, 6, 9, 3, 7

11)