  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
**Inorder Traversal**

**D C B E A F G**

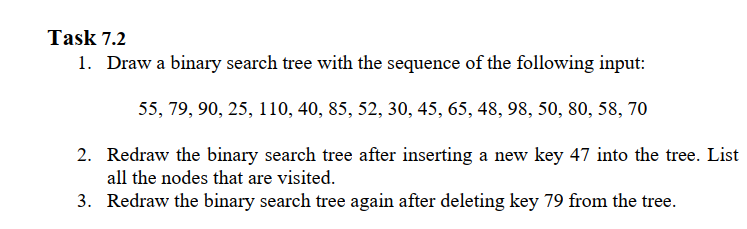
**Pre-Order Traversal**

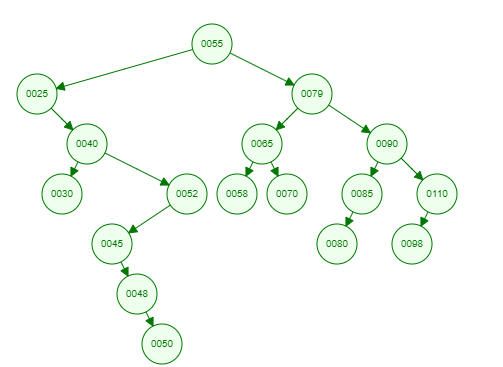
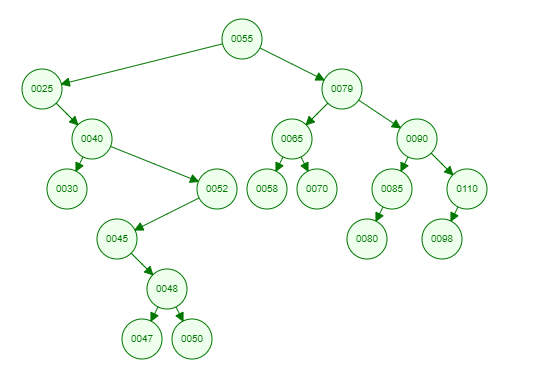
**A B C D E F G**

**Post-Order Traversal**

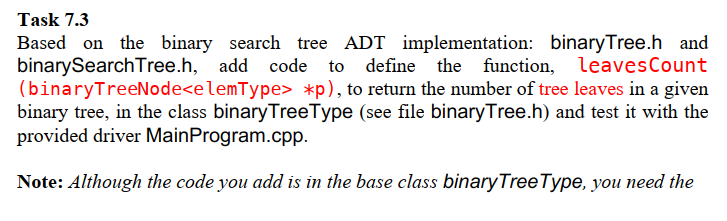
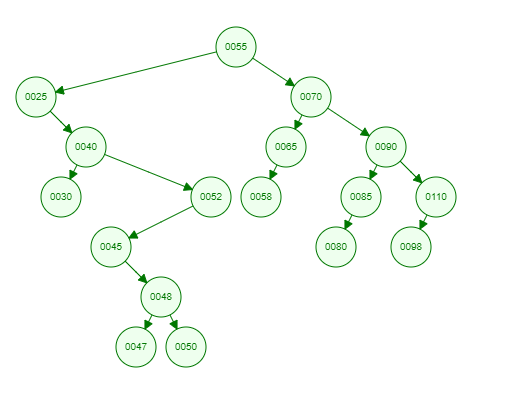
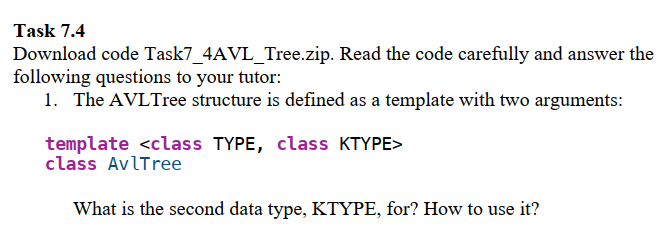
**D C E B G F A**

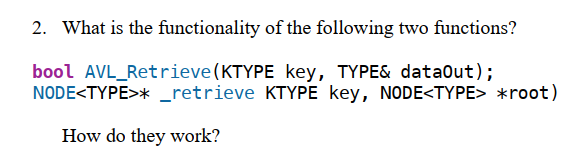
7.2



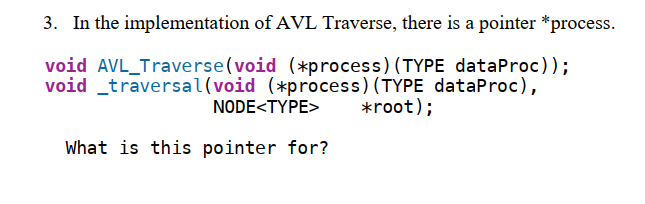


Nodes Visited: 55,25,40,52,45,48

After deleting 79:  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
The second type represents the key data type in the structure, this is different from the original type which is used to represent the structure. It’s used to retrieve the data, to determine if it is contained within the tree [if you’re using storage with a structure that has two data types, and the one is used to identify as a key within the structure].



Retrieves the data associated with the key specified by the user requesting information. This is connected to the public “AVL\_RETRIEVE” function; which takes a key value, and sends this plus the root to the \_retrieve function; this is a recursive function which returns the a pointer to the node; if not, it returns false.



It is a function pointer, this function processes the data specified by the user. It is a void function which has one formal parameter, in this case of type structure which is specified by the user; the process may modify the data or collect information [dependent on the design of the program]; in this particular case, the function conducts an operation on the data that is contained in the node. This is connected to the public AVL\_traverse function.  
  
