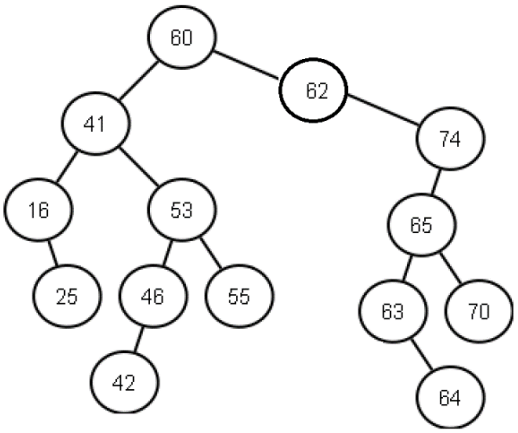
**CSCI-311 Challenge Problem 3 (50 points) Binary Search Tree**

**Problem 1.** Given a binary search tree of integers (positive and negative), BST, and an integer ***val***, find the previous node (i.e. in-order predecessor) and return the value of that node. Nodes in the BST do not have links to parents.

**Requirements:**

1. Your program (functions) must run in O(log(n))-time (if a BST is balanced, then height of the tree is O(log(n))), where *n* is the size of the given BST.
2. Your public function must be called ***previous*** and take one parameter, the given integer, and return an integer, the value of the previous node:
   int ***previous***(int val);
3. In all cases where *previous* node does not exist (BST is empty, the given node is the leftmost node of the given BST, the given value has not been found in the given BST), return INT_MIN (defined in <climits>).

**Example:**




On Figure to the left, given a node 63, your program will return 62 (the previous node of 63 if the nodes were listed using in-order traversal).

If the given node is 74, then previous node is 70.

If the given node is 16, then there are no previous node (16 is the smallest in the tree).

**Submission:**

Submit *bst.h* and *bst.cpp* to *challengeProblem3* on **turnin.**