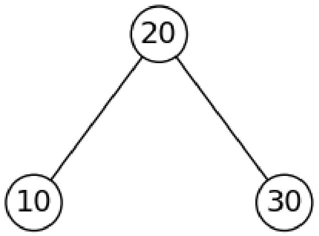
CS2302 - Data Structures

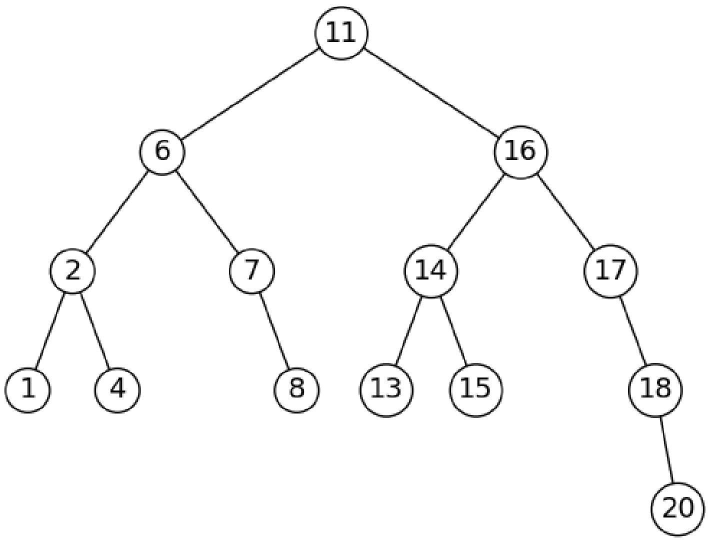
Spring 2020

Exam # 2 - Binary Search Trees

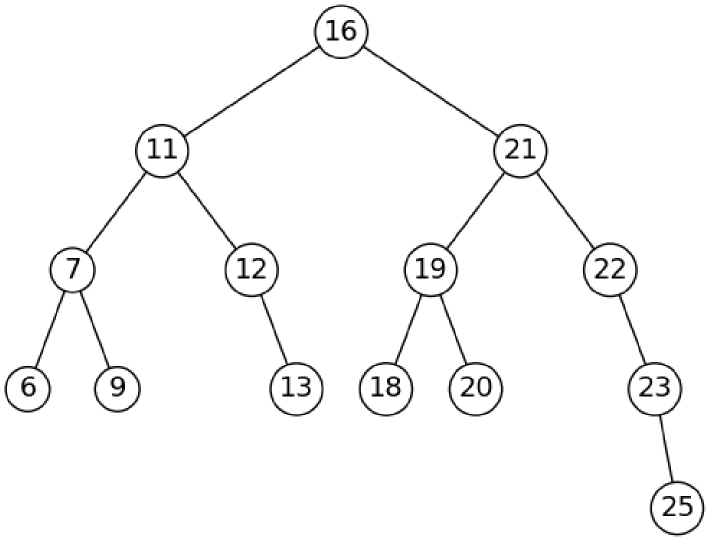
1. The function *bst_3nodes(L)* is supposed to receive a sorted list L of length 3 and build and return a balanced binary search tree containing the elements in L. For example, if $L = [10, 20, 30]$, the function is supposed to build and return the tree in the figure. However, the function does not work. Fix it so it produces the right results.



2. Write the function *has_depth(T,d)* that receives a binary search tree T and an integer d and determines if T has any nodes that have depth d. For example, if T is the tree in the figure, *has_depth(T,1)* should return *True*, since nodes 6 and 16 have depth 1, *has_depth(T,4)* should return *True*, since node 20 has depth 4, and *has_depth(T,5)* should return *False*.



3. Write the function *add_n(T,n)* that receives a binary search tree T and an integer n and adds n to every data item in the tree. For example, if T is the tree in the figure above, after executing *add_n(T,5)* T should be the tree below.



4. A path from the root to a node in a binary search tree can be encoded by a string of characters 'L' and 'R', where 'L' means 'go left' and 'R' means 'go right'. For example, in the tree from question 2, the path from the root to node 14 would be encoded by the string 'RL' (go right (16), go left (14)), the path to 1 would be encoded by the string 'LLL', and the path to the root would be encoded by '' (the empty string). Write the function *get_path(T,k)* that receives a binary search tree T and an integer k and returns a string describing the path one needs to go from the root of T to the node that contains k. If k is not in the tree, your function must return None.