CS2302 - Data Structures

Spring 2020

Exercise - Binary Search Trees

1. Write the function *printSmaller(t,k)* that receives a reference to the root of a binary search tree and an integer k and prints all the items in the tree that are less than k.

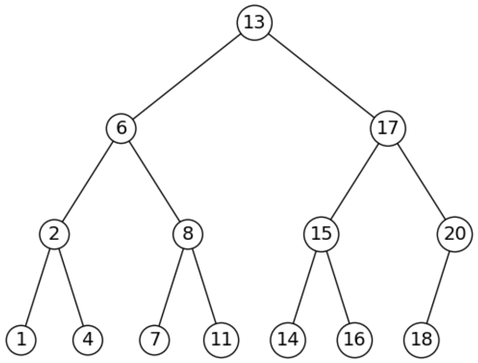
2. Write the function *printLeaves(t)* that receives a reference to the root of a binary search tree t and prints all the items in the tree that are stored in leaf nodes.

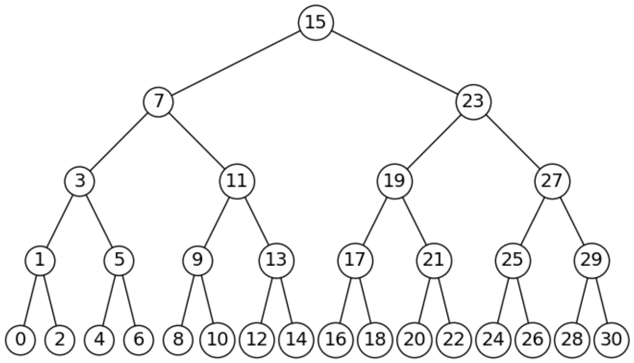
3. Write the function *atDepthD(t,d)* that receives a reference to the root of a binary search tree t and an integer d and returns a list of the items in the tree that are stored at depth d in the tree (recall that the root has depth 0, its children have depth 1, and so on).

4. Write the function *depthOfK(t,k)* that receives a reference to the root of a binary search tree t and an integer k and returns the depth of the node that contains k, or -1 if k is not in the tree.

5. Write the function *tree2List(t)* that receives a reference to the root of a binary search tree t and returns a sorted list containing the elements in the tree.

6. Write the function *list2Tree(L)* that receives a sorted list L and builds and returns a balanced (as much as possible) binary search tree containing the elements of L. The figures below were generated by the main program in the starting code.


13
6
17
2
8
15
20
1
4
7
11
14
16
18


15
7
23
3
11
19
27
1
5
9
13
17
21
25
29
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30