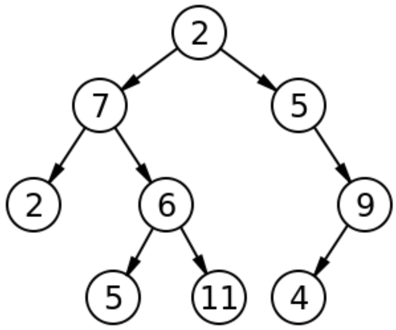
CHAPTER 6 – PROBLEMS (Part 1)

Problem 1. See the binary tree T below. Answer the questions:



a) Is T a complete binary tree? Is T a full binary tree?
b) What is the degree of T?
c) How many leaves are there? Write down the leaf nodes.
d) How many internal nodes are there? Write down the internal nodes.
e) Is T a binary search tree? Why
f) What is the height of T?
g) Write down the nodes at level 1, level 2.
h) Write down the path from root node to node 11.

Problem 2. Draw a binary tree T of height = 4 in these cases:
a) T is complete but not full.
b) T is full but not complete.
c) T is a perfect binary tree.

Problem 3. Given a full binary tree T. Let N is the total number of nodes in T, L is the total leaves and I is the total internal nodes + root node. Show that:
a) L = I + 1
b) N = 2I + 1
c) I = (N – 1)/2
d) L = (N + 1)/2
e) N = 2L – 1
f) I = L – 1

Problem 4. For the set of keys {1, 4, 5, 10, 16, 17, 21}, draw binary search trees of height 2, 3, 4, 5, and 6.

Problem 5. What is the difference between the binary-search-tree property and the min-heap property? Can the min-heap property be used to print out the keys of an *n*-node tree in sorted order in O(*n*) time? Explain how or why not.

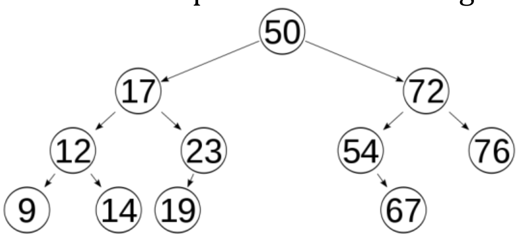
Problem 6. Suppose that we have numbers between 1 and 1000 in a binary search tree and want to search for the number 363. Which of the following sequences could not be the sequence of nodes examined?

a) 2, 252, 401, 398, 330, 344, 397, 363.
b) 924, 220, 911, 244, 898, 258, 362, 363.
c) 925, 202, 911, 240, 912, 245, 363.
d) 2, 399, 387, 219, 266, 382, 381, 278, 363.
e) 935, 278, 347, 621, 299, 392, 358, 363.

Problem 7. Write recursive versions of the functions TREE-MINIMUM and TREE-MAXIMUM in the slide.

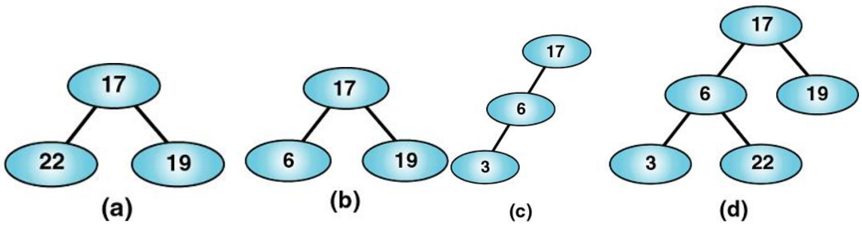
Problem 8. Professor Bunyan thinks he has discovered a remarkable property of binary search trees. Suppose that the search for key k in a binary search tree ends up in a leaf. Consider three sets: A, the keys to the left of the search path; B, the keys on the search path; and C, the keys to the right of the search path. Professor Bunyan claims that any three keys $a \in A, b \in B, c \in C$ must satisfy $a \leq b \leq c$. Give a smallest possible counterexample to the professor's claim.

Problem 9. Write down the nodes on the path of the following tree walks



a) Pre-order tree walk
b) In-order tree walk
c) Post-order tree walk

Problem 10. Which of the following trees are binary search tree:



Problem 11. Build a binary search tree with the keys: (draw the tree after each insert/delete step)
a) 8, 3, 5, 2, 20, 11, 30, 9, 18, 4. Delete the nodes: 5, 20.
b) 5, 7, 10, 12, 9, 8, 3, 1, 4. Delete the nodes 3, 10, 5.