Datastructuren

September 28, 2009

1 Assignments

1. Construct all Binary Search Trees (BSTs) T on $S = \{1, 2, 3\}$.
2. Is the binary tree in Figure 2 a binary *search* tree?
3. Redistribute the keys of the tree in Figure 2 in such a way that the form/structure of this tree stays the same and at the same time satisfies the BST property on the set $E = \{3, 5, 8, 10, 12, 13, 15, 16, 17\}$. Denote this tree by T_1.
4. Insert into the tree T_1 of Problem 3 the keys $4, 7$, and 14 .
5. Let B be a BST. Show that for a vertex of B with two children for the order relation on the set of keys E of B the following holds:
   a the successor does not have a left child,
   b the predecessor does not have a right child
6. Delete key 13 in the BST obtained in Problem 4.
7. Show that in the increasing order sequence of the keys of a BST B for any two vertices x and y: y is the successor of x if and only if $y = \min B_r(x)$ or $x = \max B_l(y)$. (Here $B_r(x)$ denotes the right subtree of B rooted in the vertex x, similarly $B_l(y)$ is the left subtree of B rooted in the vertex y. Furthermore, $\min B_r(x)$ denotes the vertex whose key is the smallest in $B_r(x)$, and similarly $\max B_l(y)$ is the vertex whose key is the largest in $B_l(y)$.)
8. Write a function which computes the smallest (resp. largest) vertex (i.e., the vertex whose key is the smallest resp. largest) in a BST. What is the complexity of this function?
9. Propose an iterative procedure/function for inserting a key into a BST. What is its complexity?
10. Write a recursively defined procedure/function for inserting a key into a BST which has the complexity as the iteratively specified version in Problem 9.
11. Write an iterative function which deletes the successor of a vertex having a right child. The function needs to return the deleted vertex.

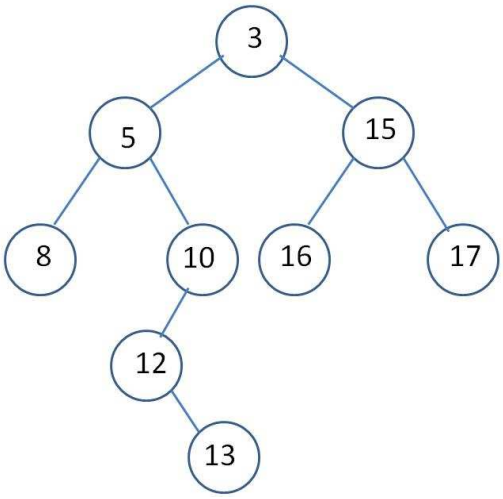

Figure 1: A Binary Search Tree

12. Propose an iterative procedure which deletes a key in a BST. What is its complexity?