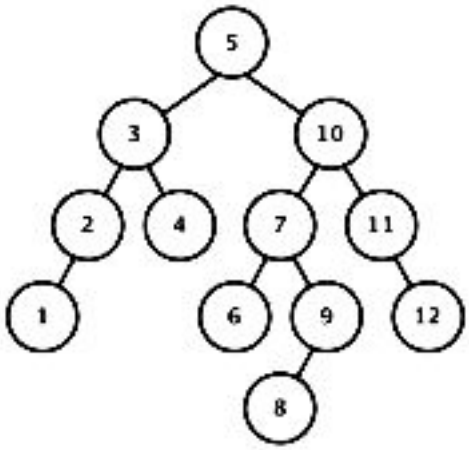
Algorithm – I CS21003
Tutorial 5
September 1st, 2017

1. We are given a set of n distinct elements and an unlabeled binary tree with n nodes. In how many ways can we populate the tree with the given set so that it becomes a binary search tree?
2. Suppose we have a balanced binary search tree T holding n numbers. We are given two numbers L and H and wish to sum up all the numbers in T that lie between L and H. Suppose there are m such numbers in T. Find the upper bound on the time to compute the sum.
3. How many distinct binary search trees can be created out of 4 distinct keys?
4. Devise an algorithm to construct a BST from a preorder traversal sequence in O(n) time, where n is the number of nodes. Ex: {10,5,2,7,45,50}
5. You are given a sequence of integers $a_1, a_2, \ldots, a_n$ in an array. You need to decide whether inserting these integers in that sequence leads to a height of n−1 of the binary search tree. Propose a worst-case O(n)-time algorithm to solve the problem. Note that if you actually build the tree, you end up in a $\Theta(n^2)$ running time in the worst case.
6. The vertex set of a binary tree T on eight nodes is {a, b, c, d, e, f, g, h}. The inorder listing of the vertices of T is bfdcgeha, and the postorder listing is fdghecba. Reconstruct the tree T. Explain the relevant steps.
7. Prove or disprove: For all integers h > 0, any AVL tree of height h + 1 contains strictly more nodes than any AVL tree of height h.
8. Consider an alternative balanced BSTs, called size-balance BSTs with the property that each node can have an imbalance of atmost 1. The imbalance of a node is defined as the absolute difference in the number of descendants of the left and right subtrees. Give an algorithm that takes as input a pointer to some node x in a BST and rebalances the subtree rooted at x. Your algorithm should run in $\Theta(l)$ time, where l is the number of descendants of x. Further, suppose that imbalance ratio of a node x is defined as the (imbalance of x / size of the tree rooted at x). Show that a BST in which each node has an imbalance ratio of atmost 0.5 has a height of O(logn).

9. Given the following AVL Tree:



   a) Draw the resulting BST after 5 is removed, but before rebalancing takes place.
   b) Draw the resulting BST after final rebalancing takes place.