Assignment 12

**Section 1: Pen-and-paper Exercises**

1. Given a BST and a positive integer k, find the k\_th smallest element in the BST.

For example, in the following BST, if k = 3, then output should be 10, and if k = 5, then output should be 14.



Assume the tree is balanced, and the tree height is O(log n). Design an O(log n) algorithm to solve this problem.

(i) describe the idea behind your algorithm in English (2 points);

(ii) provide pseudocode (5 points);

(iii) analyze its running time (3 points).

**Note: We will discuss this problem in class.**

**Note: Full credit (10 points) will be awarded for an algorithm in O(log n) time. Algorithms slower will be scored out of 5 points.**

Answer:

Solution 1: Call in-order traverse to transform the tree to a sorted array, then find the k\_th smallest element in the array – O(n) time algorithm

Solution 2: Add an extra attribute to each node: size, to represent the size of the subtree rooted at that node. Since in BST, leftsub tree data <= root data < rightsub tree data,

if the left subtree size == k-1

then root is the k\_th smallest node

or if the left subtree size > k-1

we will find the k\_th smallest node in the left subtree,

or if the left subtree size < k-1

we will find k-1-left.size smallest node in the right subtree

Runtime: O(log n)